

# A virtual rodent predicts the structure of neural activity across behaviours



<https://doi.org/10.1038/s41586-024-07633-4>

Received: 16 March 2023

Accepted: 30 May 2024

Published online: 11 June 2024

 Check for updates

Diego Aldarondo<sup>1,4</sup>, Josh Merel<sup>2,4</sup>, Jesse D. Marshall<sup>1,5</sup>, Leonard Hasenclever<sup>2</sup>, Ugne Klibaite<sup>1</sup>, Amanda Gellis<sup>1</sup>, Yuval Tassa<sup>2</sup>, Greg Wayne<sup>2</sup>, Matthew Botvinick<sup>2,3</sup> & Bence P. Ölveczky<sup>1</sup>

Animals have exquisite control of their bodies, allowing them to perform a diverse range of behaviours. How such control is implemented by the brain, however, remains unclear. Advancing our understanding requires models that can relate principles of control to the structure of neural activity in behaving animals. Here, to facilitate this, we built a ‘virtual rodent’, in which an artificial neural network actuates a biomechanically realistic model of the rat<sup>1</sup> in a physics simulator<sup>2</sup>. We used deep reinforcement learning<sup>3–5</sup> to train the virtual agent to imitate the behaviour of freely moving rats, thus allowing us to compare neural activity recorded in real rats to the network activity of a virtual rodent mimicking their behaviour. We found that neural activity in the sensorimotor striatum and motor cortex was better predicted by the virtual rodent’s network activity than by any features of the real rat’s movements, consistent with both regions implementing inverse dynamics<sup>6</sup>. Furthermore, the network’s latent variability predicted the structure of neural variability across behaviours and afforded robustness in a way consistent with the minimal intervention principle of optimal feedback control<sup>7</sup>. These results demonstrate how physical simulation of biomechanically realistic virtual animals can help interpret the structure of neural activity across behaviour and relate it to theoretical principles of motor control.

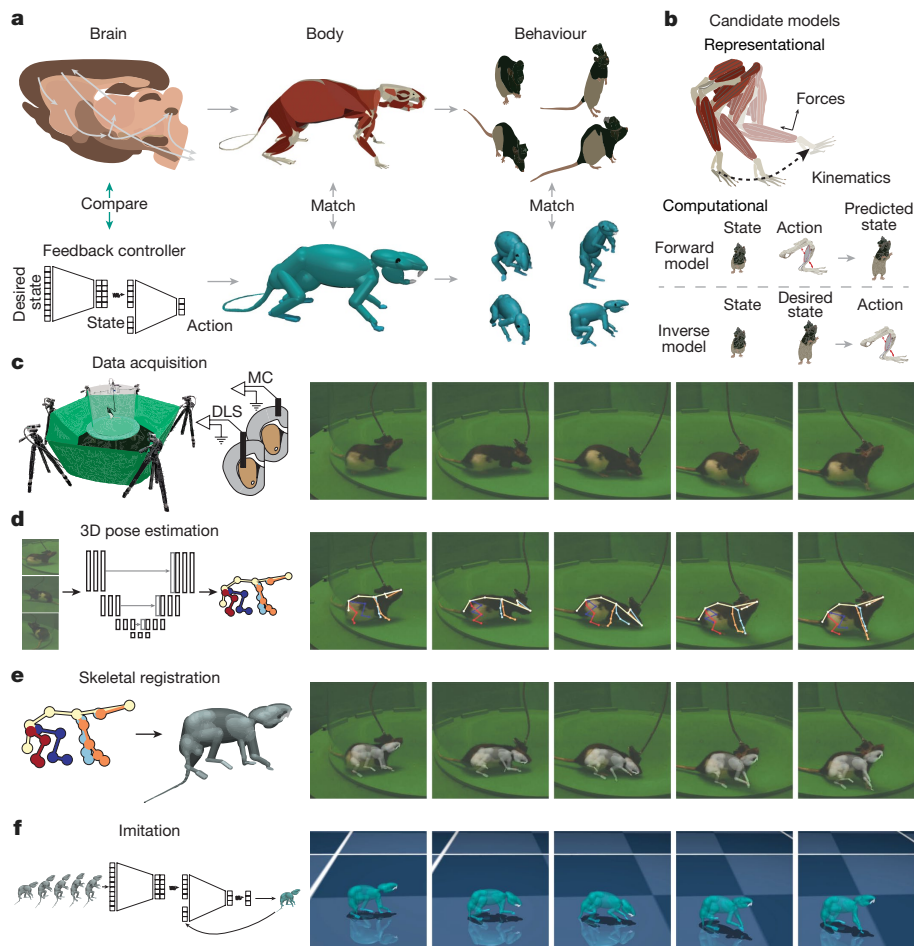
Humans and animals control their bodies with an ease and efficiency that has been difficult to emulate in engineered systems. This lack of computational analogues has hindered progress in motor neuroscience, as neural activity in the motor system is only rarely interpreted relative to models that causally generate complex, naturalistic movement<sup>8–11</sup>. In lieu of such generative models, neuroscientists have tried to infer motor system function by relating neural activity in relevant brain areas to measurable features of movement, such as the kinematics and dynamics of different body parts<sup>12–15</sup>. This is problematic because movement features are inherently correlated through physics, and the representational models based on them can only describe behaviour, not causally generate it<sup>8,16</sup>. Here, we propose an alternative approach: to infer computational principles of biological motor control by relating neural activity in motor regions to models that implement their hypothesized functions and replicate the movements and behaviours of real animals (Fig. 1a,b).

To enable this line of enquiry and probe its utility, we developed a virtual rodent by training artificial neural networks (ANNs) controlling a biomechanically realistic model of the rat to reproduce natural behaviours of real rats. This allowed us to relate neural activity recorded from real animals to the network activity of the virtual rodent performing the same behaviours. Analogous approaches have proven successful in relating the structure of neural activity to computational functions

in other domains, including vision<sup>17–19</sup>, audition<sup>20</sup>, olfaction<sup>21,22</sup>, thermosensation<sup>23</sup>, perceptual discrimination<sup>24</sup>, facial perception<sup>25</sup> and navigation<sup>26,27</sup>. However, there have been relatively few attempts to similarly model the neural control of movement, and those that did mainly probed how artificial controllers resemble neural activity during specific motor tasks or across a limited range of behaviours or effectors. Regardless, these pioneering efforts demonstrated the capacity of simple brain-inspired controllers to reproduce animal locomotion<sup>10,28,29</sup>, showed how biomechanics can influence neural representations of movement<sup>9</sup> and showed similarities between the representations of movement in artificial and biological neural networks<sup>8,30,31</sup>.

Modelling the neural control of diverse, natural behaviours is a larger undertaking with several unique challenges. Importantly, because animals evolved to skilfully control their bodies to solve challenges in complex environments<sup>32</sup>, our models should control biomechanically realistic bodies in closed loop with physically realistic environments<sup>7</sup>. Furthermore, because animals express a diverse range of species-typical behaviours, our models should be able to replicate these<sup>33</sup>. Finally, our models should demonstrate robustness to neural noise and other sources of variability inherent to biological control systems<sup>7,34</sup>. Modelling the neural control of movement at this degree of richness and realism has been hampered by a scarcity of high-fidelity three-dimensional (3D) kinematic measurements, tools to physically

<sup>1</sup>Department of Organismic and Evolutionary Biology and Center for Brain Science, Harvard University, Cambridge, MA, USA. <sup>2</sup>DeepMind, Google, London, UK. <sup>3</sup>Gatsby Computational Neuroscience Unit, University College London, London, UK. <sup>4</sup>Present address: Fauna Robotics, New York, NY, USA. <sup>5</sup>Present address: Reality Labs, Meta, New York, NY, USA. ✉e-mail: diegoaldarondo@gmail.com; olveczky@fas.harvard.edu



**Fig. 1 | Comparing biological and artificial control across the behavioural repertoire with MIMIC.** **a**, To compare neural activity in behaving animals to computational functions in control, we trained ANNs actuating a biomechanical model of the rat to imitate the behaviour of real rats. **b**, Top, representational approaches in neuroscience interpret neural activity in relation to measurable features of movement. Bottom, computational approaches, in contrast, can relate neural activity to specific control functions, such as internal models. **c–f**, The MIMIC pipeline. **c**, Left, schematic of experimental apparatus for behavioural and electrophysiological recording. A tetrode array recorded

electrical activity of neurons in DLS or MC. Right, example images taken during a walking sequence. **d**, Left, schematic of the DANNCE pose estimation pipeline. Multiview images were processed by a U-net to produce keypoint estimates. Right, walking sequence with overlaid keypoint estimates. **e**, Left, we registered a skeletal model of the rat to the keypoints in each frame using STAC. Right, walking sequence with overlaid skeletal registration. **f**, Left, we trained an ANN to actuate the biomechanical model in MuJoCo to imitate the reference trajectories. Right, walking sequence simulated in MuJoCo.

simulate animal bodies and methods to build agents that replicate the diversity of animal behaviour.

To overcome these challenges, we developed a processing pipeline called MIMIC (motor imitation and control) (Fig. 1c–f and Supplementary Video 1). MIMIC leverages 3D animal pose estimation<sup>35</sup> and an actuable skeletal model<sup>1</sup> amenable to simulation in MuJoCo<sup>2</sup>, a physics engine, to build a virtual rodent that can imitate natural behaviour under realistic constraints. Specifically, MIMIC uses deep reinforcement learning<sup>3,4</sup> to train ANNs to implement an inverse dynamics model, a function that specifies the actions (that is, joint torques) required to achieve a desired state (that is, body configuration) given the current state. We used the ANNs to control a biomechanical model of the rat, training it to imitate the movements of real rats across their behavioural repertoire. This allowed us to directly compare neural activity in freely moving animals to the activations of inverse dynamics models enacting the same behaviours.

We used this approach to interpret neural activity in the sensorimotor striatum (dorsolateral striatum (DLS) in rodents) and motor cortex (MC) of rats, two hierarchically distinct structures of the mammalian motor system for which the neural representations of natural behaviours have previously been described<sup>36–39</sup>. We found that the structure

of neural activity across behaviours was better predicted by the virtual rodent's network activity than any kinematic or dynamic feature of movement in the recorded rat, consistent with a role for both regions in implementing inverse dynamics. Furthermore, by perturbing the network's latent variability, we found that it structures action variability to achieve robust control across a diverse repertoire of behaviour in a manner consistent with theoretical principles of optimal feedback control<sup>7</sup>. Furthermore, the network activity was predictive of the structure of neural variability across behaviours, indicating that the brain structures variability in accordance with these principles.

To compare an artificial control system to a real brain producing natural behaviours requires measuring the full-body kinematics and neural activity of real animals. To this end, we recorded the behaviour of freely moving rats in a circular arena with an array of six cameras while measuring neural activity from the DLS or MC (DLS: three animals, 353.5 h, 1,249 neurons; MC: three animals, 253.5 h, 843 neurons) with custom 128-channel tetrode drives (Fig. 1c and Extended Data Fig. 1). To infer full-body kinematics from the videos, we tracked the 3D position of 23 anatomical landmarks (keypoints) on the animal using DANNCE<sup>35</sup> (Fig. 1d, Extended Data Fig. 2a–c and Supplementary Video 2). We used a feature extraction, embedding and clustering

approach to identify discrete behaviours from kinematic data, as described previously<sup>40–43</sup>. To enable physical simulation in MuJoCo<sup>2</sup>, we registered a skeletal model of the rat<sup>1</sup> with 74 degrees of freedom (DoF) (38 controllable DoF) to the keypoints using a custom implementation of the simultaneous tracking and calibration (STAC)<sup>44</sup> algorithm (Fig. 1e, Extended Data Fig. 2d–f and Supplementary Video 3). We next compiled a diverse catalogue of behavioural motifs (847 5 s snippets) spanning the behavioural repertoire of the rat to provide training data for our ANN controllers.

Controlling a complex body to perform diverse natural behaviours requires a remarkable degree of flexibility in the underlying control system. Biological control systems are widely believed to achieve such flexibility by implementing internal models: that is, neural computations that approximate the complex dynamics of the body. Here, we focus on the simplest feedback controller that uses an internal model to recapitulate the behavioural repertoire of the rat (Supplementary Discussion 1). This minimal controller takes as inputs the current state of the body, its desired future state and uses an internal model called an inverse dynamics model to estimate the action required to achieve the desired future state given the current state<sup>4,6</sup>. Despite its relatively simple formulation, building a single controller that replicates diverse behaviours while controlling a complex body is a challenging task for which performant methods have only recently been developed<sup>3–5,45</sup>.

Therefore, to build virtual rodents that imitate real animal behaviour, we trained ANNs to implement inverse dynamics models using deep reinforcement learning as in recent work (Fig. 1f)<sup>3,4</sup>. The networks accepted as input a reference trajectory of the real animal's future movements and combined a compressed representation of the reference trajectory with the current state of the body to generate an action, thus implementing an inverse dynamics model (Fig. 2a). For ease of discussion, we refer to the subnetwork that encodes the reference trajectory as the 'encoder' and the remainder of the network as the 'decoder'. The state vector was defined as the joint angular position and velocity of the virtual rodent's full-body pose, as well as simulated measurements from inertial and force sensors. The reference trajectory was defined as the states (excluding the inertial and force sensors) visited by the real rat in the immediate future (ranging from 20 to 200 ms), expressed relative to the current state of the virtual rodent's body. The action was defined as torques at 38 actuators (joints) along the body. The networks operated over short timescales to generate actions that moved the virtual rodent in the simulated environment, running at 50 Hz in a sliding-window fashion to imitate arbitrarily long bouts of behaviour. To study how different network architectures and hyperparameters affected imitation performance, we varied the decoder architecture, regularization of the latent encoding, presence of autoregression, definition of the action and reference trajectory duration. During training, the states visited by the virtual rodents were compared to the reference trajectory of the animal being imitated. This allowed us to calculate the reward at each frame using several objectives related to different kinematic and dynamic features of movement (Methods). Through trial and error, the networks learned to produce actions that moved the body of the virtual rodent in ways that matched the real animal's movements (Supplementary Video 1).

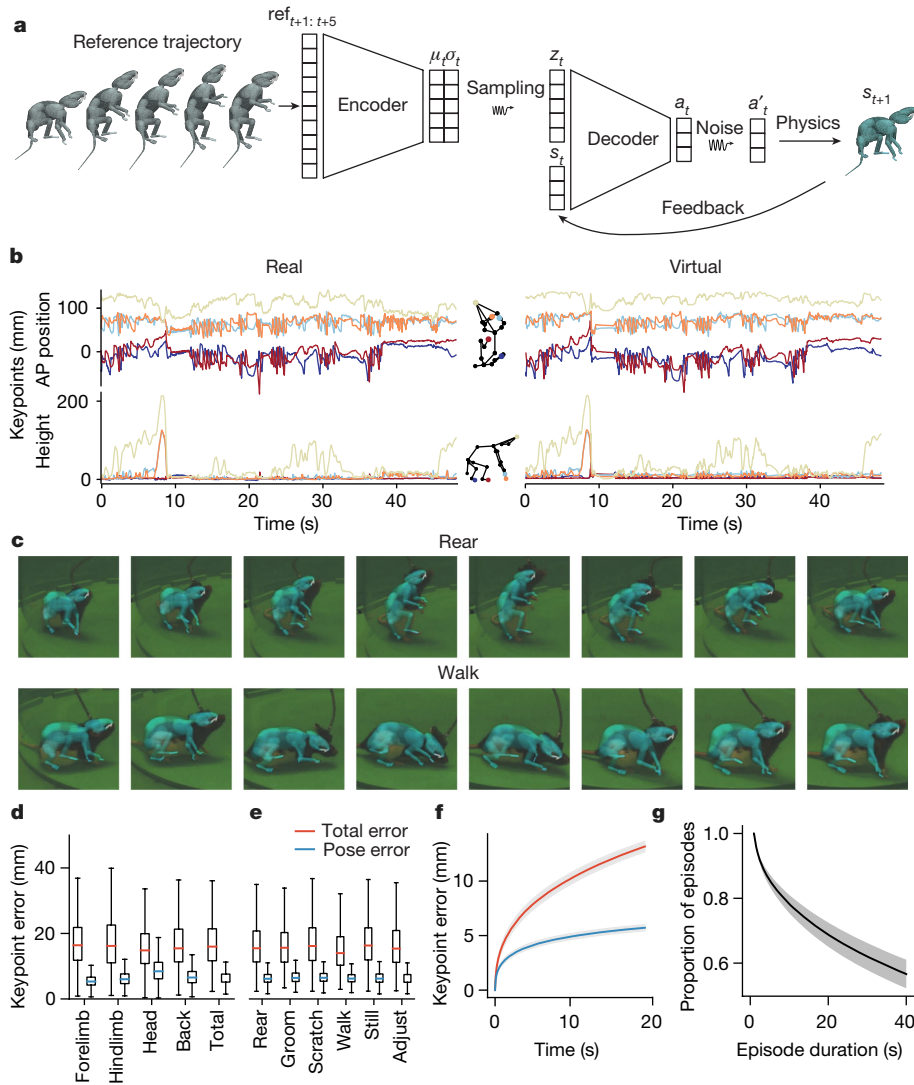
Controlling a high DoF body to imitate diverse animal movements is a challenging task for which the performance and generalization of artificial agents has only recently been characterized<sup>3–5,45</sup>. Remarkably, not only did the virtual rodent reliably and faithfully replicate movements in the training set, but the ANN controllers also generalized to held-out movements (Fig. 2b–e and Extended Data Fig. 3). This success in imitating unseen examples allowed us to evaluate the virtual rodent over the entirety of our dataset. To do so efficiently, we divided the 607 h dataset into contiguous 50 s chunks and ran the networks over all chunks in parallel. We found that all trained networks were capable of faithful imitation, but networks with recurrent decoders outperformed other architectures (Extended Data Fig. 3a,b), particularly

during slower movements (Extended Data Fig. 3c,d). For these networks, most of the deviations from the real rat's kinematics could be attributed to accumulation of error in the centre of mass (CoM) over time (Fig. 2d–f). To mitigate this, we implemented a termination and reset condition that was triggered when the virtual rodents deviated excessively from the reference trajectory (Methods). We used this to derive a measure of imitation robustness<sup>46</sup> by analysing the distribution of durations between resets, which we refer to as episode durations. Regardless of the specific ANN implementation, the virtual rodent showed remarkable robustness, imitating long bouts of behaviour without termination (Fig. 2g and Extended Data Fig. 3b). Given the short-timescale nature of the inverse dynamics models, we wondered whether providing the networks with more context about the upcoming movements would result in more robust control. For the most performant architectures, increasing the length of the reference trajectory resulted in models with greater robustness at the expense of imitation performance (Extended Data Fig. 3g–i), indicating a tradeoff between robustness and imitation fidelity when selecting the duration of the reference trajectory.

Having models that faithfully imitate natural behaviours of real rats allowed us to compare neural activity in real animals to the activations of a virtual rodent performing the same behaviours (Fig. 3a). To compare the dynamics of real and virtual control systems, we performed encoding analysis and representational similarity analysis, established methods that allowed us to probe the correspondences at the levels of both single-neuron activity and population activity structure. To establish a baseline and a point of reference, we estimated the extent to which measurable or inferable features of behaviour (representational models) relating to the kinematics and dynamics of movement (Fig. 1b) could predict the activity of putative single units (20 ms bins) in held-out data using Poisson generalized linear models (GLMs). Consistent with previous reports<sup>36,39,47</sup>, the most predictive representational feature was pose (Fig. 3b,c), with the activity of individual neurons being best predicted by the kinematics of different body parts (Extended Data Fig. 4a–d).

We next compared the predictivity of the inverse dynamics models against representational models. Although we focus on a network drawn from the most performant class of models, namely those with recurrent decoders, we note that all architectures exhibited qualitatively similar results. Because the virtual rodent produced behaviours that deviated slightly from those of real rats (Fig. 2d,e), our inverse dynamics model started at a disadvantage relative to the representational models, which were referenced to the real rat's movements. Despite this handicap, we found that the inverse dynamics model predicted the activity in both brain regions significantly better than any representational model, with the best results coming from the first layer of the decoder (Fig. 3b–d and Extended Data Figs. 4e–h and 10). We observed similar results across striatal cell types (Methods and Extended Data Fig. 5). To estimate the temporal relationships between neural activity, kinematics and our inverse dynamics model, we trained GLMs using different temporal offsets between the predictors and neural activity. Most neurons in DLS and MC were premotor, meaning that their neural activity was best predicted by future kinematics and concurrent activations of the inverse dynamics model (Extended Data Fig. 6).

To analyse the structure of population activity in MC and DLS across behaviours and assess the degree to which it is captured by representational models and inverse dynamics models, we performed representational similarity analysis (RSA)<sup>48</sup>. For our purposes, this involved quantifying how different model features were structured across behaviours using a representational dissimilarity matrix (RDM) and comparing the RDMs generated from representational features or activations of the inverse dynamics models with those generated from neural population activity in DLS and MC. For all features, we computed RDMs by calculating the average vector for each behaviour and computing the pairwise distance between these vectors using the



**Fig. 2 | Training artificial agents to imitate rat behaviour with MIMIC.** **a**, We train a virtual rodent to imitate the 3D whole-body movements of real rats in MuJoCo with deep reinforcement learning (Methods). All networks implement an inverse dynamics model that produces the actions required to realize a reference trajectory given the current state. All simulated data in this figure are derived from models with LSTM decoders. **b**, Left, keypoint trajectories of the real rat. Right, model-derived keypoint trajectories of the virtual rat imitating the real rat's behaviour (top, anterior–posterior (AP) axis; bottom, height from the floor). **c**, Example sequences of a rat performing different behaviours. Overlays rendered in MuJoCo depict the imitated movements. **d**, **e**, Imitation on held-out data is accurate for all body parts (**d**) and across different behaviours (**e**). The total error is the average Euclidean distance between the model and anatomical keypoints, whereas the pose error indicates the Euclidean distance

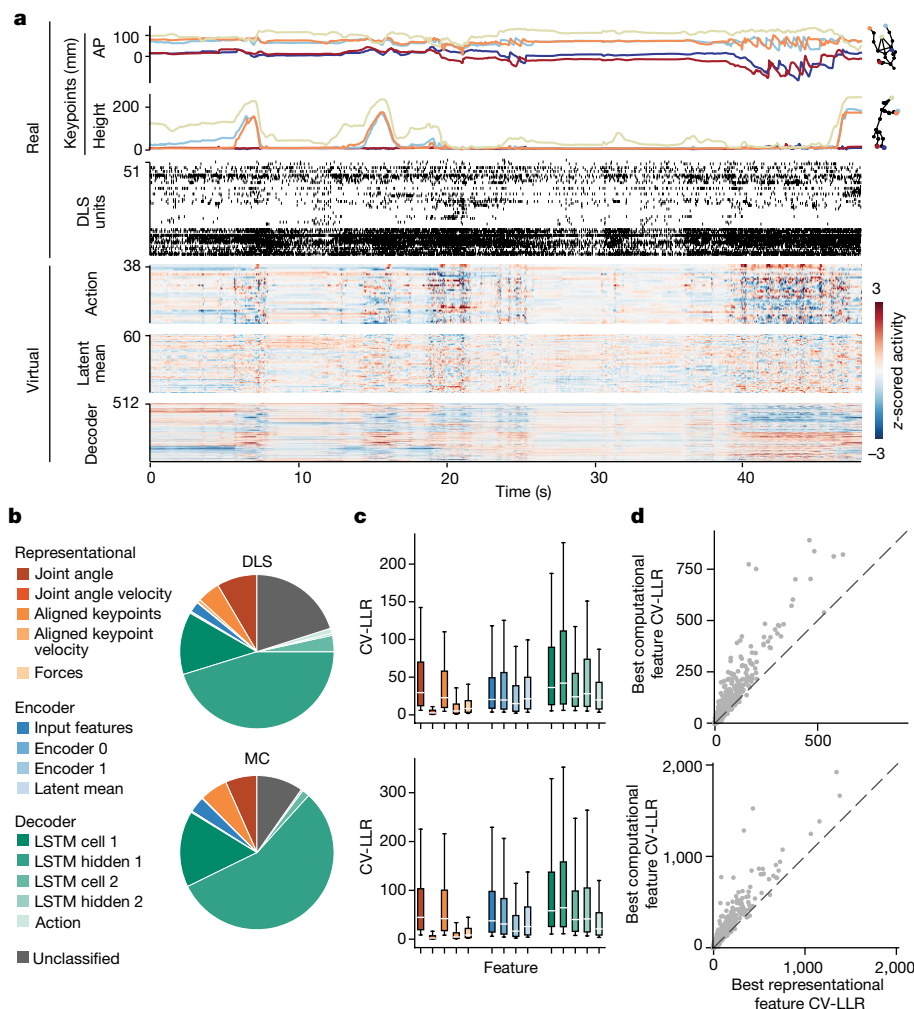
up to a Procrustes transformation without scaling. Box centres indicate median, box limits indicate interquartile range, box whiskers indicate the maximum or minimum values up to 1.5 times the interquartile range from the box limits. **b–e** feature data from a model with a recurrent decoder and a Kullback–Liebler regularization of  $1 \times 10^{-4}$ . **f**, Accumulation of error as a function of time from episode initiation. Deviations from the reference trajectory accumulate over time, with drift in the position of the centre of mass (CoM) accounting for much of the total error. **g**, The proportion of episodes exceeding a given duration. Shaded regions indicate the standard error of the mean across all models with LSTM decoders. **d–g** include data from 283 h sessions, with four sessions drawn from each of seven animals. ref, reference trajectory;  $\mu$ , latent mean;  $\sigma$ , latent variability;  $z$ , latent sample;  $s$ , state;  $a$ , action.

cross-validated Mahalanobis distance (Methods). Although we focused on the most performant network, we note that all networks exhibited qualitatively similar results. Individual neurons in DLS and MC were preferentially tuned to specific behavioural categories, resulting in RDMs that reflect the population activity structure across behaviours (Fig. 4a,b). We found that the neural population activity RDMs of both DLS and MC were more similar to the inverse dynamics model RDMs than those of the representational models (Fig. 4c–e). Moreover, when comparing across networks, we found that the similarity between RDMs constructed from inverse dynamics models and neural activity in DLS and MC was strongly correlated with the imitation performance and robustness of the network (Fig. 4f–i). This indicates that more performant models exhibit representations more similar to those of both DLS

and MC, consistent with previous reports comparing neural activity with task-optimized neural networks<sup>17,20</sup>.

To verify that the increased predictivity of inverse dynamics models relative to representational models was a result of learning the dynamics of a realistic body, we changed the body to see if it affected the fidelity of behavioural imitation and neural predictivity of our models. In a 'mass scaling' experiment, we trained the virtual rodent to control bodies with total masses that varied from half to twice the standard mass. In a 'relative head scaling' experiment, we trained it to control bodies where the mass of the head relative to the rest of the body varied from half to twice the standard ratio while maintaining the same total mass. These subtle modifications to the body model frequently resulted in policies with degraded imitation performance (Extended





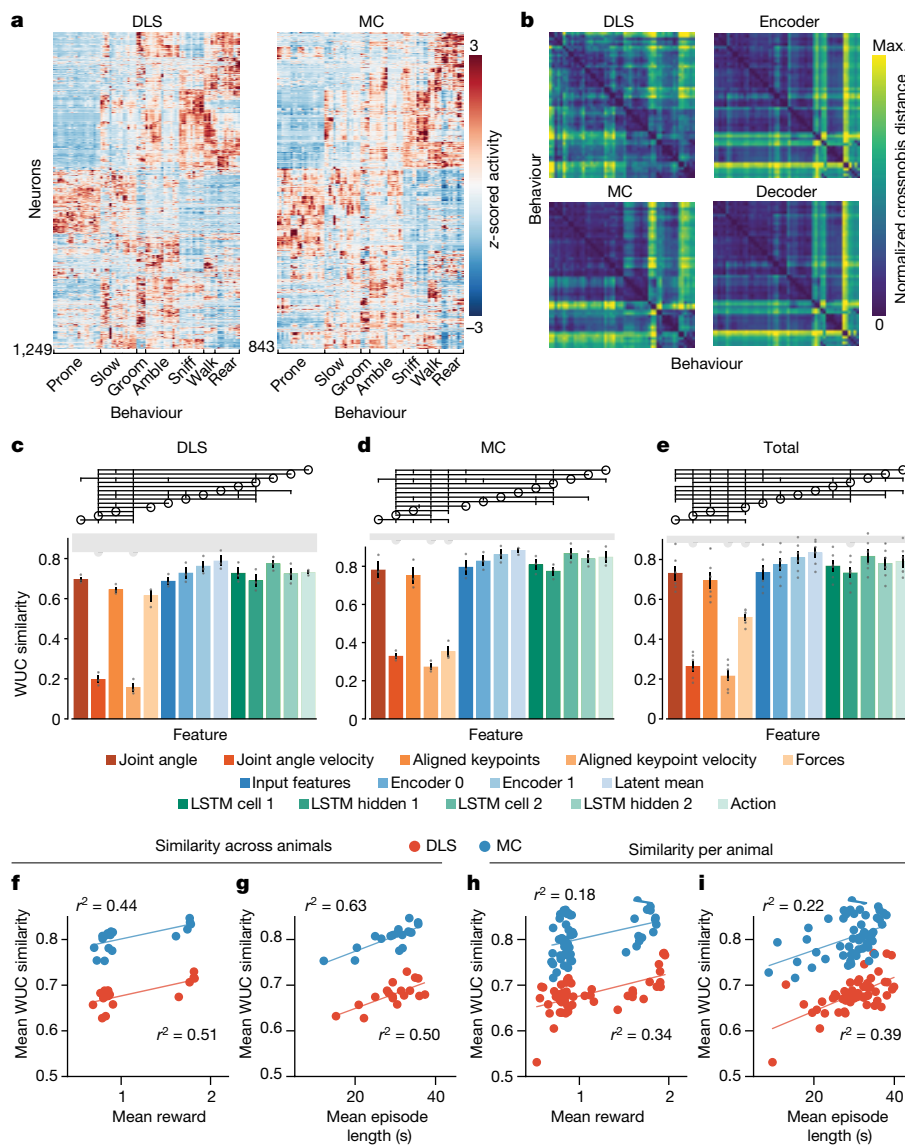
**Fig. 3 | Neural activity in DLS and MC is best predicted by an inverse dynamics model.** **a**, MIMIC enables comparisons of neural activity to measurable features of behaviour and ANN controller activations across a diverse range of behaviours. Top, aligned keypoint trajectories and spike rasters for DLS neurons over a 50 s clip of held-out behaviour. Bottom, z-scored activations of the artificial neurons forming the model’s action layer, latent mean and LSTM cell layer 1 when imitating the same clip. The depicted network features an LSTM decoder and a Kullback–Liebler regularization coefficient of  $1 \times 10^{-4}$ . **b**, Proportion of neurons in DLS and MC best predicted by each feature class. **c**, Box plots showing the distribution of cross-validated log-likelihood

ratios (CV-LLR) of GLMs trained to predict spike counts using different feature classes relative to mean firing-rate models. Data include neurons significantly predicted by each GLM (Benjamini–Hochberg corrected Wilcoxon signed-rank test,  $\alpha = 0.05$ ) from a total of  $N = 732$  neurons in DLS and 769 neurons in MC. White lines indicate the median, boxes the interquartile range and whiskers the tenth and 90th percentiles. **d**, Comparing predictions from the best computational and representational features for each neuron. GLMs based on the inverse dynamics models outperform those based on representational features for most classified neurons in both DLS and MC ( $P < 0.001$ , one-sided permutation test).

Data Figs. 7a,b and 8a,b). They also reduced overall putative single-unit predictivity from features of many inverse dynamics models (Extended Data Figs. 7c,d and 8c,d) and occasionally reduced the representational similarity to neural activity (Extended Data Figs. 7e,f and 8e,f). These results show that subtle changes to the body model can affect both the virtual rodent’s behaviour and its neural predictivity.

We next studied how the predictivity of our inverse dynamics models compared to that of ANNs implementing other control functions. To test this, we used data from the most performant inverse dynamics model (Methods) to train ANNs by means of supervised learning<sup>49,50</sup> to implement a forward model and a sequential forecasting model (Extended Data Fig. 9a–c). Neither model could predict putative single-unit activity in MC or DLS more accurately than the inverse dynamics model (Extended Data Fig. 10a,b). Similarly, neither model could predict the representational similarity structure of MC and DLS as well as the inverse dynamics model (Extended Data Fig. 9d–f), consistent with these brain areas reflecting computations associated with inverse dynamics.

In addition to imitating animal behaviour and predicting the structure of neural activity, simulated controllers allow us to study control processes that are difficult to access experimentally. A long-standing question that can be uniquely studied in this way relates to how movement variability is shaped by the nervous system. It has been widely observed that animals structure movement variability differently depending on the task, with variability preferentially quenched along task-relevant dimensions in accordance with the minimal intervention principle<sup>7,51,52</sup>. In the context of optimal feedback control, such ‘structured variability’ is thought to result from regularizations of the cost functions associated with movement generation<sup>52</sup>, such as the minimization of jerk<sup>53</sup> or energy expenditure. However, with the notable exception of signal-dependent noise<sup>54</sup>, how neural activity in biological control networks shapes variability in motor output remains largely unexplored (although see ref. 55). To address this, we leveraged the stochastic nature of our inverse dynamics models to study whether and how its ‘neural’ variability structured action variability.



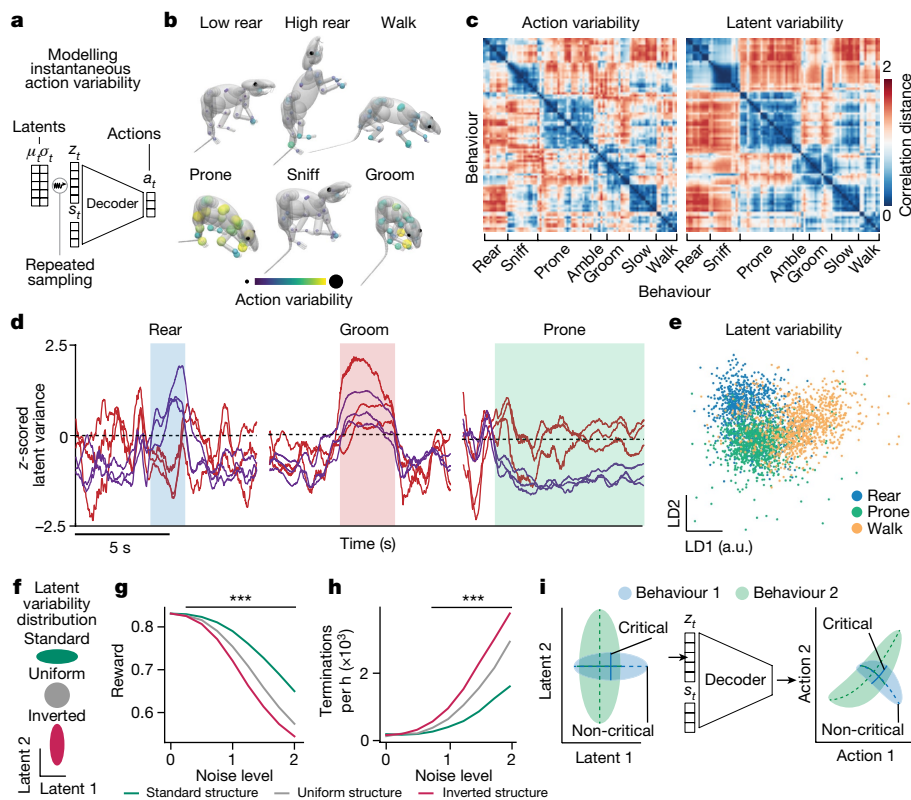
**Fig. 4 | The representational structure of neural populations in DLS and MC across behaviours resembles that of an inverse model. a**, Average normalized firing rate for single units in DLS and MC as a function of behaviour. **b**, Average RDMs for neural activity in DLS and MC and the average of layers in the encoder and decoder. Row and column indices are equal across RDMs and sorted by means of hierarchical clustering on the average neural activity RDM across all animals. **c–e**, Cross-subject average of whitened unbiased cosine (WUC), similarity between RDMs of different computational and representational models and neural activity. Layers of the inverse dynamics model predict the dissimilarity structure of neural activity in DLS (panel **c**) and MC (panel **d**) better than representational models (panel **e**). Error bars indicate s.e.m. and circles and dew drops indicate significant differences from the noise ceiling and zero (Bonferroni corrected,  $\alpha = 0.05$ , one-sided  $t$ -test). Grey bars indicate

the estimated noise ceiling of the true model. Open circles indicate the comparison model; downward ticks on the wings extending from the comparison model indicate significant differences between models (Benjamini–Hochberg corrected, false discovery rate  $\alpha = 0.05$ , one-sided  $t$ -test). Points indicate individual animals ( $N = 3$  individuals in **c** and **d**,  $N = 6$  individuals in **e**). **f**, Comparing average imitation reward and the mean WUC similarity with DLS or MC neural activity on held-out data for all networks. The average WUC similarity is the average similarity of all network layers relative to neural activity for a given network. Each point denotes a single network across all animals for a given brain region. **g**, Comparison of average WUC similarity and the average episode length for all networks. **h, i**, Same as **f** (**h**) and **g** (**i**) except each point denotes a single network–animal pair. Max., maximum.

To probe the relationship between ‘neural’ variability and action variability, we focused on two components of the network: the latent variability, a part of the latent encoding that parametrizes the variability of a 60-dimensional Gaussian distribution, and the action the network outputs. We used the generative nature of the latent encoding to relate latent variability at a given timepoint (‘instantaneous’ latent variability) to the variability of the distribution of actions that emerge from repeated resampling of this latent encoding (‘instantaneous’ action variability; Fig. 5a). We use the phrase instantaneous variability to differentiate from other types of variability, such as trial-to-trial variability or temporal variability.

Importantly, these quantities can only be directly accessed through simulation.

To determine whether the virtual rodent’s actions exhibited structured variability, we estimated the instantaneous action variability at each timepoint (Fig. 5a) and averaged across behavioural categories. As in biological controllers, the structure of variability across the model’s actuators showed a strong dependence on the behaviour (that is, the task) being performed (Fig. 5b). However, unlike in biological controllers, signal-dependent noise cannot contribute to this structured variability as none of the sources of variability in the network were signal-dependent by construction. Consistent with action variability



**Fig. 5 | Stochastic controllers regulate motor variability as a function of behaviour by changing latent variability.** **a**, We estimate instantaneous action variability as the standard deviation of the set of actions obtained by resampling the latent space 50 times at every step. To avoid long-term temporal dependencies, simulations in this figure use a torque-actuated controller with an MLP decoder and a Kullback–Liebler regularization coefficient of  $1 \times 10^{-3}$ . **b**, Action variability differs as a function of behaviour ( $P < 0.001$ , one-sided permutation test; see Methods). Each sphere corresponds to a single actuator; its colour and size indicate its normalized action variability during the designated behaviour. **c**, RDMs of action variability and latent variability across behaviours. **d**, Trajectories of six latent dimensions along which variability was differentially regulated across behaviour. Shaded regions indicate times in which corresponding behaviours were performed. Dimensions coloured in red increase variability during grooms, whereas dimensions coloured in purple

increase variability during rears and grooms and quench variability during prone. **e**, Scatter plot depicting the latent variability at single time points plotted on the first two linear discriminants for three behavioural categories. The population latent variability discriminates behaviours ( $P < 0.001$ , one-sided permutation test; LD1 and LD2 denote linear discriminants 1 and 2 respectively, see Methods). **f**, Schematic depicting changes to the structure of latent variability (see text). **g, h**, The deviations from normal variability structure reduce the model’s robustness to noise ( $P < 0.001$ , one-sided Welch’s  $t$ -test) (**g**) and termination rate ( $P < 0.001$ , one-sided Chi-squared test) (**h**). Lines indicate significant differences between conditions. **i**, The latent variability is differentially shaped as a function of behaviour to structure action variability in accordance with the minimal intervention principle. a.u., arbitrary units.

being controlled by the network’s latent variability, we found that their dissimilarity structures across behaviours were similar (Fig. 5c), with individual latent dimensions expanding or contracting their variability as the virtual rodent performed different behaviours (Fig. 5d). Indeed, the behavioural dependence was so strong that the latent variability alone was sufficient to identify the behaviour enacted at any given time (Fig. 5e). To relate the latent variability structure across behaviour in the virtual rodent with the neural variability structure of real animals, we compared RDMs derived from the latent variability and the temporal variability of neural activity evaluated over a 1 s moving window (Methods). Intriguingly, the latent variability structure resembled that of the temporal variability of neural activity across behaviours (Extended Data Fig. 9g–i), indicating that the inverse dynamics models predict not only the structure of neural activity but also its variability.

To determine whether the structure of the virtual rodent’s latent variability afforded robustness in accordance with the minimal intervention principle, we changed it in two different ways. We made the variability across all dimensions of the latent encoding uniform and, in a different simulation, inverted the variability: that is, quenching it in dimensions with normally high variability and vice versa (Methods and Fig. 5f). These deviations from the virtual rodent’s learned variability structure

resulted in poorer imitation and more frequent failures at equal noise levels (Fig. 5f–h), consistent with the system’s variability structure obeying the minimal intervention principle<sup>7</sup>. To be clear, we do not suggest that the latent variability itself improves performance or robustness. In fact, models with stronger latent regularization and thus greater latent variability performed slightly worse in terms of imitation reward and robustness on the testing set (Extended Data Fig. 3e,f). Instead, these results, coupled with the training objective (Supplementary Discussion 2), show that the virtual rodent adaptively shapes latent variability to increase robustness according to behavioural demands (Fig. 5i), affording robustness in the face of unquenchable noise. This structured variability emerges solely from latent variable compression, indicating a link between mechanisms for robustness and generalizability (see also refs. 25,56).

## Discussion

How the brain achieves robust and flexible control of complex bodies has remained a puzzle in large measure because of the lack of expressivity in our models and their reductionist nature. Here, we address these limitations by taking a holistic approach to sensorimotor control that

emphasizes embodiment, sensory feedback, stochastic control and diverse behaviour. Our approach reflects a belief that motor system function cannot be understood independent of the body it evolved to control or the behaviours it evolved to produce (Fig. 1a).

To demonstrate the utility of this approach, we developed a virtual rodent in which a fully configurable and transparent ANN controls a biomechanically realistic model of a rat in a physics simulator (Supplementary Discussion 3). In constructing such a system, we needed to balance tractability, expressivity and biological realism at the level of both the biomechanical plant and its controller. In this work, we opted for the simplest model that could recapitulate the behavioural repertoire of the rat and predict the structure of neural activity in the brain across behaviour. The result was a plant with point-torque actuation and a controller implementing inverse dynamics. Our simulations show that this level of model abstraction, which notably omits muscular actuation, is already sufficient to achieve our objectives. As we and others extend these biomechanical models to include whole-body musculoskeletal actuation<sup>57</sup>, it will be interesting to probe the degree to which such increased biomechanical realism further informs our understanding of the neural control of movement.

To train the virtual rodents to replicate spontaneous behaviours of real rats (Fig. 2), we developed an imitation learning pipeline, MIMIC (Fig. 1c–f). We found that the virtual rodents generalized to unseen movements across the entirety of our dataset with high fidelity and robustness (Fig. 2 and Extended Data Fig. 3). In comparing the network activations to neural recordings in real rats performing the same behaviours, we found that our model explained the structure of recorded neural activity across a wide range of naturalistic behaviours better than any traditional representational model (Figs. 3 and 4). Because the virtual rodent's ANN implements an inverse dynamics model, the observation that its network activations predict single-unit and population neural activity in DLS and MC more accurately than measurable features of movement or alternative control functions is consistent with these regions taking part in implementing inverse dynamics.

We believe that the improvements in predictivity relative to representational models result from incorporating bodily dynamics, including the influences of gravity, friction, inertia and interactions between body parts. Note, however, that one should ostensibly be able to find a combination of measurable representational features that predicts neural activity as well as our models. In fact, this is what our models do. They learn a nonlinear function, parametrized by a neural network, that transforms the kinematics of desired future movements into the dynamics required to achieve those kinematics. The network does this by encoding the physical realities of bodily control in its weights; that is, it learns an inverse dynamics model. There are at least two main advantages of our approach relative to approaches based on representation. First, the models we train are causal—they are sufficient to physically reproduce the behaviour of interest as opposed to merely describing it. Second, they place the emphasis on identifying the functions that brain regions implement as opposed to merely describing the flow of information.

Previous work in brain–machine interfaces<sup>58</sup> and oculomotor control<sup>59</sup> has similarly related neural activity in the motor system to inverse dynamics models. Our work extends these findings to the domain of full-body control and across a diverse behavioural repertoire. We note that a neural code consistent with inverse dynamics could reflect and support other processes, including motor learning<sup>60</sup> or even different control functions. However, in our experiments, models trained to implement forward dynamics and sequential forecasting did not fare as well in predicting neural activity structure (Extended Data Fig. 9a–f), although we note that these controls may differ from internal models (such as state estimation, forward dynamics models and so on) implemented in a composite controller<sup>6</sup>. In future work, a more comprehensive understanding of the relationship between control functions and neural activity structure could be achieved by comparing the network

activations of such composite controllers to neural recordings in brain regions believed to implement different internal models.

Although noise is inherent to biological control, how the nervous system deals with it to ensure robustness and flexibility remains unclear<sup>61</sup>. The minimal intervention principle speaks to this, explaining that controllers quench movement variability along dimensions relevant to performance<sup>7,51</sup>. By leveraging stochastic ANN controllers (Fig. 2a), the virtual rodent allowed us to study the relationship between network variability and variability in motor output. Intriguingly, we found that the virtual rodent 'brain' regulated its latent variability to control action variability in accordance with the minimal intervention principle<sup>7</sup> (Fig. 5). The structure of variability emerged from training the network to balance latent variable compression, implemented to support generalization and the ability to faithfully imitate over the training set (Supplementary Discussion 2). Thus, managing the tradeoff between latent variable compression and motor performance<sup>25,56</sup> may structure neural variability in ways that are distinct from previously hypothesized mechanisms like signal-dependent noise<sup>54</sup> or energy constraints<sup>7</sup>. Together, these results show a link between a computational mechanism for generalization (latent variable compression) and one for robustness in control (structured variability). That the virtual rodent's latent variability predicts the structure of neural variability across behaviours in DLS and MC (Extended Data Fig. 9g–i) further indicates that the brain may structure neural variability in accordance with these principles.

More generally, our results demonstrate how artificial controllers actuating biomechanically realistic models of animals can help uncover the computational principles implemented in neural circuits controlling complex behaviour. We believe the potential of this approach is significant and untapped. Virtual animals trained to behave like their real counterparts could provide a platform for virtual neuroscience to model how neural activity and behaviour are influenced by variables like feedback delays, latent variability and body morphology that would otherwise be difficult or impossible to experimentally deduce. Inverse dynamics models trained to reproduce diverse and realistic behaviours could also be re-used as low-level modules to promote naturalistic movement in neural networks trained to autonomously perform tasks, including those common in neuroscience research<sup>3,4</sup>. Similarly, because the ANNs controlling the virtual rodent are fully configurable, future iterations could aim to implement brain-inspired network architectures to improve performance and interpretability and probe the roles of specific circuit motifs and neural mechanisms in behaviourally relevant computations.

## Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at <https://doi.org/10.1038/s41586-024-07633-4>.

1. Merel, J. et al. Deep neuroethology of a virtual rodent. In *Proc. 8th International Conference on Learning Representations* 11686–11705 (ICLR, 2020).
2. Todorov, E., Erez, T. & Tassa, Y. MuJoCo: a physics engine for model-based control. In *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* 5026–5033 (IEEE, 2012).
3. Hasenclever, L., Pardo, F., Hadsell, R., Heess, N. & Merel, J. CoMic: complementary task learning & mimicry for reusable skills. In *Proc. 37th International Conference on Machine Learning* (eds Daumé, H. & Singh, A.) 4105–4115 (PMLR, 2020).
4. Merel, J. et al. Neural probabilistic motor primitives for humanoid control. In *Proc. 7th International Conference on Learning Representations* (ICLR, 2019).
5. Peng, X. B., Abbeel, P., Levine, S. & van de Panne, M. DeepMimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.* **37**, 1–14 (2018).
6. Jordan, M. I. In *Handbook of Perception and Action*, Vol. 2 (ed. Heuer, H.) Ch. 2 (Academic Press, 1996).
7. Todorov, E. & Jordan, M. I. Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.* **5**, 1226–1235 (2002).



8. Todorov, E. Direct cortical control of muscle activation in voluntary arm movements: a model. *Nat. Neurosci.* **3**, 391–398 (2000).
9. Lillcrap, T. P. & Scott, S. H. Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics. *Neuron* **77**, 168–179 (2013).
10. Ijspeert, A. J., Crespi, A., Ryzcko, D. & Cabelguen, J.-M. From swimming to walking with a salamander robot driven by a spinal cord model. *Science* **315**, 1416–1420 (2007).
11. Kalidindi, H. T. et al. Rotational dynamics in motor cortex are consistent with a feedback controller. *eLife* **10**, e67256 (2021).
12. Georgopoulos, A. P., Kalaska, J. F., Caminiti, R. & Massey, J. T. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J. Neurosci.* **2**, 1527–1537 (1982).
13. Evars, E. V. Relation of pyramidal tract activity to force exerted during voluntary movement. *J. Neurophysiol.* **31**, 14–27 (1968).
14. Ashe, J. Force and the motor cortex. *Behav. Brain Res.* **87**, 255–269 (1997).
15. Kalaska, J. F. From intention to action: motor cortex and the control of reaching movements. *Adv. Exp. Med. Biol.* **629**, 139–178 (2009).
16. Churchland, M. M. & Shenoy, K. V. Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *J. Neurophysiol.* **97**, 4235–4257 (2007).
17. Yamins, D. L. K. et al. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl Acad. Sci. USA* **111**, 8619–8624 (2014).
18. Kar, K., Kubilius, J., Schmidt, K., Issa, E. B. & DiCarlo, J. J. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nat. Neurosci.* **22**, 974–983 (2019).
19. Khaligh-Razavi, S.-M. & Kriegeskorte, N. Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.* **10**, e1003915 (2014).
20. Kell, A. J. E., Yamins, D. L. K., Shook, E. N., Norman-Haignere, S. V. & McDermott, J. H. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron* **98**, 630–644 (2018).
21. Wang, P. Y., Sun, Y., Axel, R., Abbott, L. F. & Yang, G. R. Evolving the olfactory system with machine learning. *Neuron* **109**, 3879–3892 (2021).
22. Singh, S. H., van Breugel, F., Rao, R. P. N. & Brunton, B. W. Emergent behaviour and neural dynamics in artificial agents tracking odour plumes. *Nat. Mach. Intell.* **5**, 58–70 (2023).
23. Haesemeyer, M., Schier, A. F. & Engert, F. Convergent temperature representations in artificial and biological neural networks. *Neuron* **103**, 1123–1134.e6 (2019).
24. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013).
25. Higgins, I. et al. Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nat. Commun.* **12**, 6456 (2021).
26. Banino, A. et al. Vector-based navigation using grid-like representations in artificial agents. *Nature* **557**, 429–433 (2018).
27. Cueva, C. J. & Wei, X.-X. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. In *Proc. 6th International Conference on Learning Representations (ICLR)*, 2018.
28. Grillner, S. et al. in *Progress in Brain Research*, Vol. 165 (eds Cisek, P. et al.) 221–234 (Elsevier, 2007).
29. Knüsel, J., Crespi, A., Cabelguen, J.-M., Ijspeert, A. J. & Ryzcko, D. Reproducing five motor behaviors in a salamander robot with virtual muscles and a distributed CPG controller regulated by drive signals and proprioceptive feedback. *Front. Neurobot.* **14**, 604426 (2020).
30. Michaels, J. A., Schaffelhofer, S., Agudelo-Toro, A. & Scherberger, H. A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proc. Natl Acad. Sci. USA* **117**, 32124–32135 (2020).
31. Sussillo, D., Churchland, M. M., Kaufman, M. T. & Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* **18**, 1025–1033 (2015).
32. Chiel, H. J. & Beer, R. D. The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends Neurosci.* **20**, 553–557 (1997).
33. Scott, S. H. & Loeb, G. E. The computation of position sense from spindles in mono- and multiarticular muscles. *J. Neurosci.* **14**, 7529–7540 (1994).
34. Latash, M. L., Scholz, J. P. & Schöner, G. Motor control strategies revealed in the structure of motor variability. *Exerc. Sport Sci. Rev.* **30**, 26–31 (2002).
35. Dunn, T. W. et al. Geometric deep learning enables 3D kinematic profiling across species and environments. *Nat. Methods* **18**, 564–573 (2021).
36. Mimica, B., Dunn, B. A., Tombaz, T., Bojja, V. P. T. N. C. S. & Whitlock, J. R. Efficient cortical coding of 3D posture in freely behaving rats. *Science* **362**, 584–589 (2018).
37. Markowitz, J. E. et al. The striatum organizes 3D behavior via moment-to-moment action selection. *Cell* **174**, 44–58.e17 (2018).
38. Klaus, A. et al. The spatiotemporal organization of the striatum encodes action space. *Neuron* **95**, 1171–1180.e7 (2017).
39. Mimica, B. et al. Behavioral decomposition reveals rich encoding structure employed across neocortex in rats. *Nat. Commun.* **14**, 3947 (2023).
40. Marshall, J. D. et al. Continuous whole-body 3D kinematic recordings across the rodent behavioral repertoire. *Neuron* **109**, 420–437.e8 (2021).
41. Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface* **11**, 20140672 (2014).
42. Klibaite, U. et al. Deep phenotyping reveals movement phenotypes in mouse neurodevelopmental models. *Mol. Autism* **13**, 12 (2022).
43. Pereira, T. D. et al. Fast animal pose estimation using deep neural networks. *Nat. Methods* **16**, 117–125 (2018).
44. Wu, T., Tassa, Y., Kumar, V., Movellan, J. & Todorov, E. STAC: simultaneous tracking and calibration. In *Proc. 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* 469–476 (IEEE, 2013).
45. Peng, X. B., Ma, Z., Abbeel, P., Levine, S. & Kanazawa, A. AMP: adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph.* **40**, 1–20 (2021).
46. Fussell, L., Bergamin, K. & Holden, D. SuperTrack: motion tracking for physically simulated characters using supervised learning. *ACM Trans. Graph.* **40**, 1–13 (2021).
47. Dhawale, A. K., Wolff, S. B. E., Ko, R. & Ölveczky, B. P. The basal ganglia control the detailed kinematics of learned motor skills. *Nat. Neurosci.* **24**, 1256–1269 (2021).
48. Kriegeskorte, N., Mur, M. & Bandettini, P. Representational similarity analysis - connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* **2**, 4 (2008).
49. Jordan, M. I. & Rumelhart, D. E. Internal world models and supervised learning. In *Proc. 8th International Workshop on Machine Learning* (eds Birnbaum, L. A. & Collins, G. C.) 70–74 (Morgan Kaufmann, 1991).
50. Nagabandi, A., Kahn, G., Fearing, R. S. & Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. Preprint at <https://arxiv.org/abs/1708.02596> (2017).
51. Valero-Cuevas, F. J., Venkadesan, M. & Todorov, E. Structured variability of muscle activations supports the minimal intervention principle of motor control. *J. Neurophysiol.* **102**, 59–68 (2009).
52. Diedrichsen, J., Shadmehr, R. & Ivry, R. B. The coordination of movement: optimal feedback control and beyond. *Trends Cogn. Sci.* **14**, 31–39 (2010).
53. Flash, T. & Hogan, N. The coordination of arm movements: an experimentally confirmed mathematical model. *J. Neurosci.* **5**, 1688–1703 (1985).
54. Harris, C. M. & Wolpert, D. M. Signal-dependent noise determines motor planning. *Nature* **394**, 780–784 (1998).
55. Wolpert, D. M. Probabilistic models in human sensorimotor control. *Hum. Mov. Sci.* **26**, 511–524 (2007).
56. Lai, L. & Gershman, S. J. in *Psychology of Learning and Motivation*, Vol. 74 (ed. Federmeier, K. D.) Ch. 5 (Academic Press, 2021).
57. Ramalingasetty, S. T. et al. A whole-body musculoskeletal model of the mouse *IEEE Access* **9**, 163861–163881 (2021).
58. Golub, M., Chase, S. & Yu, B. Learning an internal dynamics model from control demonstration. In *Proc. 30th International Conference on Machine Learning* (eds Dasgupta, S. & McAllester, D.) 606–614 (PMLR, 2013).
59. Shidara, M., Kawano, K., Gomi, H. & Kawato, M. Inverse-dynamics model eye movement control by Purkinje cells in the cerebellum. *Nature* **365**, 50–52 (1993).
60. Kawai, R. et al. Motor cortex is required for learning but not for executing a motor skill. *Neuron* **86**, 800–812 (2015).
61. Faisal, A. A., Selen, L. P. J. & Wolpert, D. M. Noise in the nervous system. *Nat. Rev. Neurosci.* **9**, 292–303 (2008).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

© The Author(s), under exclusive licence to Springer Nature Limited 2024

## Methods

### Data acquisition

**Animals.** The procedures involved in the care and experimental manipulation of all animals were reviewed and approved by the Harvard Institutional Animal Care and Use Committee. Experimental subjects included seven female Long Evans rats, aged 3–12 months at the start of recording (Charles River).

**Behavioural apparatus.** Animals moved freely in a cylindrical arena 1 m in diameter elevated on a custom-made wooden table. The circular base of the cylinder was made of green high-density polyethylene cutting board. The walls were 60 cm tall and made of a 1-mm-thick clear polycarbonate sheet. The arena was surrounded by a commercial green screen to improve contrast between the animals and their surroundings. The arena was illuminated by two white light-emitting diode arrays (Genaray SP-E-500B, Impact LS-6B stands) to aid kinematic tracking. To encourage movement, three to six Cheerios were hung around the arena using pieces of string such that they were within reach of the rats when rearing.

**Videography.** Six high-speed 2 MP Basler Ace-2 Basic cameras (a2A1920-160160ucBAS) were equipped with 8 mm lenses (Lens Basler 8 mm, C23-0824-5M, 2/3", f/2.4, 5 MP) and placed surrounding the arena at regular intervals approximately 1.2 m from the centre. Cameras were stabilized with SLIK PRO 700DX tripods. All camera shutters were controlled synchronously by a 50 Hz Arduino hardware trigger by means of Phoenix Contact Sensor/actuator cables (SAC-6P-M AMS/3.0-PUR SH - 1522309). Images were transmitted by means of Basler USB 3.0 cables to an acquisition computer equipped with an Intel Core i9-9900K processor, an NVIDIA Quadro P4000, an NVIDIA Geforce GTX 1550 Super and a Samsung 970 Evo M.2 solid-state drive. We used the Campy camera acquisition software suite to encode videos from all cameras in real time<sup>35</sup>.

**Calibration.** Cameras were calibrated using tools from the MATLAB 2020b ([https://www.mathworks.com/downloads/web\\_downloads/](https://www.mathworks.com/downloads/web_downloads/)) and OpenCV-Python (v.4.4.0.46) (<https://pypi.org/project/opencv-python/#history>) camera calibration libraries. For intrinsic calibration, we used the MATLAB Single Camera Calibrator App with a checkerboard calibration pattern to estimate the camera parameters for each camera individually. For extrinsic calibration, we placed the same checkerboard used for intrinsic calibration in the centre of the arena, took a picture from all cameras, detected checkerboard corners in all images using the functions in OpenCV-Python calibration library and estimated the rotation and translation vectors for each camera using MATLAB's extrinsic calibration functions. Calibrations were checked periodically to ensure that the cameras had not been accidentally disturbed between recordings. In practice, we found the recording apparatus to be stable enough that calibrations would remain accurate for months at a time.

**Electrophysiology.** Microdrive construction and surgical procedures for tetrode implantation followed previously described protocols<sup>62</sup>, with slight modifications to accommodate 128-channel recordings. Notably, an array of 32 tetrodes was manually connected to a custom-designed headstage (made of two RHD2164 ICs from Intan Technologies), rather than the 16 tetrodes used in previous designs. All implants were in the right hemisphere. Target coordinates were 0 mm anterior–posterior, +4.25 mm mediolateral, –4 mm dorsoventral for the DLS and +1 mm anterior–posterior, +1 mm mediolateral, –1 mm dorsoventral for MC relative to bregma. MC targets were chosen to match the median recording location of MC recordings by ref. 36. In one MC implant, the target site was moved by approximately +1 mm anterior–posterior and +1 mm mediolateral to avoid a blood vessel.

We occasionally lowered the drive by approximately 80  $\mu\text{m}$ , 0–4 times over the course of the experiments. Recordings were conducted using the Intan RHX2000 acquisition software. Electrophysiological and video data were synchronized by passing the video hardware trigger signal through the acquisition field-programmable gate array (Opal Kelly XEM6010, Xilinx Spartan-6 field-programmable gate array) that interfaced with the headstage. One animal implanted in MC yielded no neurons and was thus excluded from electrophysiological analyses.

**Recording protocol.** Single-housed rats were manually placed in the arena at the beginning of a recording session and left alone and undisturbed for 2 or 3 h. All recordings were performed in the absence of experimenters in a closed room with minimal noise and began at approximately the same time every day. Animals were recorded daily for a minimum of 28 days and a maximum of 63 days. The arena floor and walls were cleaned with 70% ethanol after every recording session and allowed to dry for at least 30 min before further use. In total, the dataset spans 607 h of simultaneous electrophysiology and videography (353.5 h DLS and 253.5 h MC).

**Histology.** At the end of the experiment, we performed an electrolytic lesion of the recording site by passing a 30  $\mu\text{A}$  current through the electrodes. For two animals implanted in MC, we were unable to perform a lesion as the headstages came off unexpectedly. The location of these implants was verified on the basis of scarring caused by the implant. After lesioning, animals were euthanized (100 mg  $\text{kg}^{-1}$  ketamine; 10 mg  $\text{kg}^{-1}$  xylazine) and transcidentally perfused with 4% paraformaldehyde in 1 $\times$  phosphate buffered saline. We then extracted the brains and placed them in 4% paraformaldehyde for 2 weeks. Brains were sectioned into 80  $\mu\text{m}$  slices using a vibratome (Vibration Company Vibratome 1500 Sectioning System), and the slices were mounted on microscope slides and stained with Cresyl-Violet. We imaged the slides using an Axioscan slide scanner and localized the recording site by the electrolytic lesions.

### Data processing

**3D pose estimation.** We used DANNCE v.1.3 to estimate the 3D pose of the animal over time from multicamera images. Pose estimation with DANNCE consists of two main steps: CoM detection and DANNCE keypoint estimation.

**CoM network training.** We used Label3D<sup>35</sup> to manually label the rat CoM from multicamera images in 600 frames spanning three animals. Frames were manually selected to span the range of locations and poses animals assume when in the arena. CoM networks were trained as described previously<sup>35</sup>.

**DANNCE network training.** We again used Label3D to manually label the 3D positions of 23 keypoints along a rat's body. The dataset consisted of over 973 frames manually selected to sample a diverse range of poses from four different animals over eight different recordings. We finetuned a model previously trained to track keypoints in the Rat7M (<https://doi.org/10.6084/m9.figshare.c.5295370.v3>) dataset on our training set, as in earlier work<sup>35</sup>. Notable modifications to this procedure included two methods for data augmentation and a modified loss function. The first data augmentation method is mirror augmentation, which effectively doubles the dataset size by inverting the 3D volumes generated from multicamera images along the  $x$  axis (parallel to the ground) and swapping the 3D positions of bilaterally symmetric keypoints. The second is view augmentation, which randomly permutes the order that images from different cameras are fed into the network. Finally, we used an L1 loss function rather than the original L2 loss. We include a list of relevant DANNCE parameter specifications in Supplementary Table 2.

**Evaluation.** DANNCE performance was quantified using a dataset of 50 manually labelled frames randomly selected from a recording session that had not been included in the training set. To estimate intralabeller

# Article

variability, the same 50 frames were re-labelled by the same person one month after the initial labelling. We report the keypoint error between manual labels and DANNCE predictions up to a Procrustes transformation without scaling.

**Compute resources.** CoM and DANNCE models were trained and evaluated using computational resources in the Cannon High Performance Cluster operated by Harvard Research Computing. These included a mixture of NVIDIA hardware including GeForce GTX 1080 Ti, GeForce RTX 2080 Ti, Tesla V100, A40 and A100 Tensor Core graphics processing units.

**Skeletal model.** We previously developed a skeletal model of a rat that matches the bone lengths and mass distribution of Long Evans rats<sup>1</sup>. The model has 74 DoF and defines parent-child relationships between body parts through an acyclic tree that starts with the root (similar to the CoM) and branches to the extremities. The pose of the model consists of three Cartesian dimensions specifying the position of the root in space, four dimensions specifying the quaternion that captures the orientation of the model relative to the Cartesian reference and 67 dimensions that specify the orientations of child body parts relative to their parent's reference frame. The model has 38 controllable actuators that apply torques to specific joints. To help imitate rearing, we increased the range of motion of the ankle and toe joints to  $[-0.1, 2.0]$  and  $[-0.7, 0.87]$  radians. The model is equipped with a series of sensors, including (1) a velocimeter, (2) an accelerometer, (3) a gyroscope and (4) force, torque and touch sensors on its end effectors.

**Skeletal registration.** We used a custom implementation of STAC<sup>44</sup> to register the skeletal model to the DANNCE keypoints. Briefly, STAC uses an iterative optimization algorithm to learn a set of 3D offsets that relate different sites along the skeletal model to DANNCE keypoints (m-phase), as well as the pose of the model that best reflects the keypoints at each frame given the set of offsets (q-phase). To ensure consistent relationships between keypoints and model sites across different poses, the offsets corresponding to keypoints closest to a body part were expressed in the reference frame of the parent body part.

In the m-phase, we optimize the offsets using L-BFGS-B over a dataset of 500 frames to minimize the mean-squared error between the true keypoints and fictive keypoints derived from applying the offsets to the posed model. In the q-phase, we optimize the pose of the model using least-squares optimization over the same set of frames to minimize the same objective while keeping the offsets fixed. At each step of the pose optimization, we reposition the model and compute new positions of the fictive offsets by means of forward kinematics in MuJoCo.

As the dataset totalled 607 h of data sampled at 50 Hz, the registration algorithm needed to be efficient. To speed up the q-phase, we separately optimized the pose of different body parts rather than optimizing over the full-body pose. First we initialize the model's root position as the position of the middle spine keypoint and optimize only the seven DoF specifying the Cartesian position and quaternion of its root. We next optimize the quaternions of the root, trunk and head to match keypoints along the head and trunk of the animal. Finally, we individually optimize each limb. In subsequent frames, we initialize the model's pose using its pose in the previous frame.

For each animal, we independently estimated the offsets using the procedure described above. We accounted for differences in animal size by isometrically scaling the model by a scaling factor manually determined by means of a visual comparison of models overlain on images of the rats. We found this procedure to be more robust and faster and to produce comparable results to a direct optimization of the scaling factor when learning offsets. In practice, we found that running the iterative optimization three times produced reasonable offsets that could be used to estimate the skeletal pose in new keypoint data.

To infer the skeletal pose of an entire recording session, we ran the q-phase optimization a final time using the set of offsets learned during

training. To improve inference speed, we divided the session into contiguous 20 s chunks and ran the q-phase optimization in parallel on Harvard Research Computing's Cannon High Performance Cluster.

**Behavioural segmentation.** We automatically identified stereotyped behaviours throughout our recording sessions using an unsupervised feature extraction and clustering approach described previously<sup>40–43</sup>. We extracted a high-dimensional feature vector capturing the multi-scale dynamics of the animal's keypoints over time. The vector was composed of three types of features. The first was the height of the keypoints from the floor, smoothed using a 60-ms-median filter. The second was the keypoint velocities, estimated using the finite differences method on smoothed (100-ms-median filter) keypoint trajectories. The third was a multiscale time-frequency decomposition of the rat's pose, obtained by computing a pairwise distance matrix between keypoints for all frames in the smoothed keypoint trajectories, decomposing the matrix into its top 20 principal components and applying a continuous wavelet transform to each principal component with frequencies ranging from 0.5 to 20 Hz.

To aid in identifying diverse stereotyped behaviours, we implemented a sampling and clustering procedure. For each session, we subsampled the feature vector by a factor of 20 and embedded it into a two-dimensional space using  $t$ -distributed stochastic neighbour embedding. We next clustered the resulting space using hierarchical watershed clustering and uniformly sampled 500 samples across the clusters. Samples from each session were compiled into a single set and clustered to automatically assign behavioural categories to individual frames using  $k$ -means clustering ( $K = 100$ ). The resulting cluster centroids were then used to classify the remaining frames in the original dataset.

**Spike sorting.** For each animal, the raw neural data from all sessions was sorted using an improved implementation of Fast Automated Spike Tracker (FAST)<sup>62</sup>. Although most the sorting process remains unchanged between the implementations, there are three relevant modifications.

**Feature extraction.** We applied a  $\beta$ -distribution ( $\beta = 100$ ) weighting transform to the spike waveforms to more heavily weigh the values near the spike peak. We next spectrally decomposed the waveforms using a discrete wavelet transform with a Symlets 2 wavelet. Finally, we applied principal components analysis on the wavelet coefficients, retaining only the first ten components.

**Clustering.** We identified putative single units using ISO-split<sup>63</sup> rather than superparamagnetic clustering. We used an iso-cut threshold of 0.9, a minimum cluster size of 8, 200 initial clusters and 500 iterations per pass.

**Linking.** To sort our long-term recordings, we clustered the feature-transformed data spanning chunks of approximately 1 h using ISO-split and linked clusters across chunks using a variation of the segmentation fusion algorithm detailed in FAST. The relevant modification was using the Komolgorov-Smirnov criterion from ISO-split to link similar neurons across recording sessions.

**Criteria for unit selection.** After manual curation, we used several summary statistics to further assess the quality of putative single units. For encoding analyses, we excluded units with an isolation index<sup>62</sup> less than 0.1 and a proportion of interspike interval violations greater than 0.02. For single-unit analyses, we excluded units with average firing rates less than 0.25 Hz and a total recorded duration that failed to span the entirety of the session in which they were measured. For population analyses, we excluded units with average firing rates less than 0.05 Hz and those with total recorded durations that failed to span the entirety of the session in which they were measured (2,092 putative single units, 1,249 DLS, 843 MC). For all analyses, spike times were binned into 20 ms bins.

## Model training

**Training set.** The training data for MIMIC controllers consists of contiguous trajectories of a high-dimensional state vector describing the real animal's movement. Features in this vector were derived from the skeletal registration and are as follows: freejoint Cartesian position, root quaternion, joint quaternions, CoM, end effector Cartesian position, freejoint velocity, root quaternion velocity, joint quaternions velocity, appendage Cartesian positions, body Cartesian positions and body quaternions. Velocities were estimated with the finite differences method.

We automatically identified a collection of 5 s clips containing a wide range of behaviours spanning our behavioural embedding. We found it necessary to prioritize behaviours in which the animal was moving to prevent the model from converging to local minima where it would remain still. We visually verified the quality of each clip by removing clips in which the animal did not move or seemingly assumed physically implausible poses from errors in tracking or registration. In the end we used a dataset of 842 clips.

**Imitation task.** We used an imitation task similar to previous works on motion capture tracking<sup>5,64,65</sup> and most closely resembling CoMic<sup>3</sup>. The task has four main considerations: initialization, observations, reward function and termination condition.

**Initialization.** Episodes were initiated by randomly selecting a starting frame from the set of all frames across all clips, excluding the last ten frames from each clip. The pose of the rat model was initialized to the reference pose in the selected frame.

**Observations.** The model received as input a combination of proprioceptive information, motion and force sensors and a reference trajectory. These include the actuator activation, appendage positions, joint positions, joint velocities, accelerometer data, gyroscope data, touch sensors at the hands and feet, torque sensors at the joints, velocimeter data, tendon position, tendon velocities and the reference trajectory. The reference trajectory is defined as the set of states visited by the real animal in a short time window ranging from 20 to 200 ms in the future (most models had a time window duration of 100 ms). At each timepoint, the kinematics of the reference trajectory was represented relative to the current state of the model in Cartesian and quaternion representations. Given the short timescales of the reference trajectory, we believe our models are most appropriate for interpreting the short-timescale dynamics involved in motor control, rather than the long-term organization of behaviour.

**Reward functions.** As in previous work on motion capture tracking<sup>3,5,64</sup>, we treat the imitation objective as a combination of several rewards pertaining to different kinematic features. The rewards consist of four terms that penalize deviations between the reference trajectory and the model's kinematics and one term that regularizes actuator forces.

The first term,  $r_{\text{com}}$ , penalizes deviations in the positions of the CoM between the reference and model:

$$r_{\text{com}} = \exp(-100 \|p_{\text{com}} - p_{\text{com}}^{\text{ref}}\|^2)$$

where  $p_{\text{com}}$  and  $p_{\text{com}}^{\text{ref}}$  are the CoM positions for the model and reference, respectively. Only spatial dimensions parallel to the ground were included to avoid the ambiguity in CoM height between isometrically scaled versions of the model used for skeletal registration and the unscaled versions of the model used in training.

The second term,  $r_{\text{vel}}$ , penalizes deviations in the joint angular velocities between the reference and model:

$$r_{\text{vel}} = \exp(-.1 \|q_{\text{vel}} - q_{\text{vel}}^{\text{ref}}\|^2)$$

where  $q_{\text{vel}}$  and  $q_{\text{vel}}^{\text{ref}}$  are the joint angle velocities of the model and reference, respectively, and the difference is the quaternion difference.

The third term,  $r_{\text{app}}$ , penalizes deviations in the end effector appendage position between the reference and the model:

$$r_{\text{app}} = \exp(-400 \|p_{\text{app}} - p_{\text{app}}^{\text{ref}}\|^2)$$

where  $p_{\text{app}}$  and  $p_{\text{app}}^{\text{ref}}$  are the end effector appendage positions of the model and reference, respectively.

The fourth term,  $r_{\text{quat}}$ , penalizes deviations in the root quaternions of the model and reference:

$$r_{\text{quat}} = \exp(-2 \|q_{\text{quat}} - q_{\text{quat}}^{\text{ref}}\|^2)$$

where  $q_{\text{quat}}$  and  $q_{\text{quat}}^{\text{ref}}$  are the root quaternions of the model and reference, respectively.

The fifth term,  $r_{\text{act}}$ , regularized the actuator forces used across the agent's actuators:

$$r_{\text{act}} = -0.015 \frac{1}{D} \sum_{i=1}^D a_i^2$$

where  $D$  is the number of controllable actuators and  $a_i$  is the actuator force of the  $i$ th actuator

**Termination condition.** Episodes were automatically terminated when the model's movements substantially deviated from the reference. Specifically, episodes terminated when

$$1 - \frac{1}{\tau} (\|b_{\text{pos}} - b_{\text{pos}}^{\text{ref}}\|_1 + \|q_{\text{pos}} - q_{\text{pos}}^{\text{ref}}\|_1) < 0$$

where  $\tau$  corresponds to the termination threshold,  $b_{\text{pos}}$  and  $b_{\text{pos}}^{\text{ref}}$  correspond to the body positions of the model and reference and  $q_{\text{pos}}$  and  $q_{\text{pos}}^{\text{ref}}$  correspond to the joint angles of the model and reference, with the difference being the quaternion difference. We used a  $\tau$  value of 0.3 in all experiments.

**Training.** Models were trained using several objective value maximum a posteriori policy optimization (MO-VMPO)<sup>66</sup>. In this setting, MO-VMPO trains a single policy to balance five objectives corresponding to each of the five reward terms. The relative contribution of each objective is specified by a vector,  $\epsilon$ , with a single element per objective. We set  $\epsilon_{\text{com}} = 0.01$ ,  $\epsilon_{\text{app}} = 0.01$ ,  $\epsilon_{\text{vel}} = 0.01$ ,  $\epsilon_{\text{quat}} = 0.01$  and  $\epsilon_{\text{act}} = 0.0001$ . For all models, we used a batch size of 256, an unroll length of 20 and a discount factor of 0.95. In the MO-VMPO E-steps, we use the top 50% of advantages<sup>67</sup>. In the policy distillation step, we set the Kullback–Liebler bounds for the policy mean to 0.1 and the Kullback–Liebler bounds for the policy covariance to  $1 \times 10^{-5}$ . We initialized all Lagrange multipliers in MO-VMPO to 1, with minimum values of  $1 \times 10^{-8}$ . We used Adam<sup>68</sup> for optimization with a learning rate of  $1 \times 10^{-4}$ . Models were trained using 4,000 actors, 32 cachers and a TPUv2 chip. A typical model trained for 2–3 days.

## Model architectures

An overview of model architectures is included in Supplementary Table 3.

**Reference encoder.** All architectures featured the same reference encoder. We used the reference trajectory for the following five timesteps and proprioceptive observations at the current timestep as inputs to the reference encoder. The encoder consisted of a two-layer densely connected multilayer perceptron (MLP) with 1,024 hidden units in each layer and hyperbolic tangent activation functions, using layer norm. The final layer of the encoder produced two 60-dimensional vectors that were passed through linear activation functions to respectively parametrize the mean,  $\mu$ , and log standard deviation,  $\sigma$ , of the stochastic latent representation.

**MLP value function.** For MLP networks, the critic was composed of a two-layer MLP with 1,024 hidden units, followed by one more one-layer



# Article

MLP for each objective. It received the same inputs as the reference encoder.

**LSTM value function.** For long short-term memory (LSTM) networks, the critic was composed of a single LSTM with 512 hidden units, followed by one more one-layer MLP for each objective. It received the same inputs as the reference encoder.

**Latent regularization.** As in CoMic<sup>3</sup>, we append a further Kullback–Liebler divergence (KL) loss term to the MO-VMPO policy distillation objective that regularizes the latent embedding using a standard Gaussian prior,

$$\beta \mathbb{E}_{\pi} [KL(\pi(z_t | s_t, s_t^{\text{ref}}) || \mathcal{N}(0, I))]$$

with the scalar parameter  $\beta$  controlling the strength of the regularization,  $z_t$  indicating the latent at time  $t$ ,  $s_t$  indicating the state at time  $t$ , and  $s_t^{\text{ref}}$  indicating the state of the reference trajectory at time  $t$ . We also impose a one-step autoregressive prior, AR(1), described by

$$z_{t+1} = \alpha z_t + \sigma \epsilon, \epsilon \sim \mathcal{N}(0, I)$$

where  $\alpha$  is the contribution of the autoregressive term, and  $\sim$  indicates ‘distributed as’. For models with autoregressive priors, we use  $\alpha$  to 0.95, and for those without autoregressive priors, we set  $\alpha$  to 0.

**MLP decoder.** The MLP decoder was composed of a two-layer MLP with 1,024 hidden units.

**LSTM decoder.** The LSTM decoder was composed of two stacked LSTMs with 512 and 256 hidden units, respectively.

**Action type.** We trained models with two types of actions. The first was position-controlled action, in which model outputs denoted the desired position of each controllable actuator. The forces required to achieve those positions were then computed by means of inverse kinematics to actuate the model appropriately. The second was torque-controlled action, in which the model directly produced torques at each actuator.

**Reference trajectory duration.** In one experiment (Extended Data Fig. 3g,h), we trained five inverse dynamics models that varied in the duration of the reference trajectory (20, 40, 60, 100 or 200 ms). The models all featured torque actuation, a LSTM decoder and a Kullback–Liebler regularization coefficient of  $1 \times 10^{-4}$ .

**Body modifications.** In two separate experiments (Extended Data Figs. 7 and 8), we trained inverse dynamics models to control modified versions of the virtual rodent body. These modifications were designed to influence the dynamics of movement without requiring changes in the kinematics of movement. In a ‘mass scaling’ experiment, we uniformly scaled the masses of all body parts of the virtual rodent body from half to twice the standard mass. In a ‘relative head scaling’ experiment, we scaled the mass of the head relative to the mass of the rest of the body from half to twice the standard ratio. In both experiments, we trained inverse dynamics models with torque actuation, a LSTM decoder and a Kullback–Liebler regularization coefficient of  $1 \times 10^{-4}$  to control the different modified bodies and evaluated their performance on held-out data controlling the bodies on which they were trained.

## Model inference

**Rollout.** To evaluate the models on new data, we used the postural trajectories obtained from STAC as reference trajectories. At each frame, the model would accept its current state and the reference trajectory for the following frames and generate an action. After applying forward kinematics, this action would result in the state at the next

frame, closing the sensorimotor loop. In the initial frame, the model’s state was initialized to the state of the real animal. For the encoding and representational similarity analyses, we disabled the noise at the action periphery and the sampling noise in the stochastic latent space. For analyses of the model’s latent variability, these sources of noise remained enabled.

At each frame, we recorded physical parameters related to the model’s state, activations of several layers of the ANN controllers and the fictive reward. The physical parameters included STAC-estimated keypoints, quaternion forces experienced at all joints, quaternion positions, velocities and accelerations and the cartesian positions of all joints. The recorded ANN layers included the latent mean and log standard deviation, the latent sample, all LSTM hidden and cell states and the action.

To aid in comparing the network’s activity to neural activity, we maintained the termination condition used during training. This decision had two effects. First, it maintained that the model’s behaviour was within a reasonable range of the true behaviour. Second, it ensured that the state inputs to the model were within the distribution observed during training and thus prevented the network activity from behaving unpredictably. For all analyses, we excluded the 0.2 s preceding or following initialization or termination frames.

As this rollout process is serial and limited by the speed of the physical simulation, evaluating long sessions is time consuming. To improve inference speed, we divided all recordings into 50 s chunks and evaluated models on each chunk in parallel, using one central processing unit core per chunk.

## Alternative control models

To compare the structure of neural activity across behaviour to functions other than inverse models, we use a dataset of state–action pairs obtained from MIMIC model rollouts when imitating natural behaviour to implement forward and sequential forecasting models in ANNs using supervised learning (Extended Data Fig. 10a–f). States were parametrized by the model’s quaternion pose, whereas actions were parametrized by the model’s action. Forward models were trained to predict the sensory consequences of motor actions, transforming the state and action for the current frame into the state of the next frame. Sequential forecasting models were trained to predict future states from past states. We varied the number of frames spanning the past-state vector from one to five to test the influence of longer context while maintaining parity with the window size of the inverse models.

The encoders and decoders for both models were composed of multilayer perceptrons with three hidden layers of 1,024 units each, with leaky rectified linear unit activation functions<sup>69</sup>. All models featured  $\beta$ -weighted conditional latent bottlenecks of equal dimensionality to those of the inverse dynamics models (60), with a  $\beta$  value of 0.01. The objective was to minimize the mean-squared error of the target.

Although we believe that comparisons to models trained by means of supervision is valuable, it is possible that the representations of models trained by means of reinforcement to implement alternative control functions may differ from those trained through supervision. This question could be resolved in future work by means of the integration of several control functions into composite controllers trained by means of reinforcement.

## Encoding analyses

**Feature set.** We used Poisson GLMs with a log link function<sup>70</sup> to predict the spiking of putative single units in DLS and MC from measurable movement features, features inferred from physical simulation in MuJoCo and the activations of ANN inverse controllers. The measurable features included aligned 3D keypoint positions and velocities and joint angular positions and velocities, spanning the entire body. Dynamic features inferred from MuJoCo included the forces experienced at each joint and accelerometer, velocimeter and touch sensors. Finally, the

ANN activations included the activations of every layer of the inverse dynamics models, considered independently.

To ensure that the models were trained to predict movement-related activity rather than activity during sleeping or resting, we focused only on frames in which animals were moving. To estimate moving frames, we smoothed the keypoint velocities estimated by means of finite differences with a five-frame median filter and identified frames in which the average smoothed keypoint velocity was above a threshold of 0.07 mm per frame. We then estimated sleeping frames as the set of frames resulting from the application of 20 iterations of binary closing (binary erosion followed by binary dilation) and 500 iterations of binary opening (binary dilation followed by binary erosion) to the vector of non-moving frames.

For all features, we used data from a temporal window containing the five surrounding samples to predict the number of spikes in a given bin. In general, increasing the window size improved model predictivity up until five frames. We also trained models over a range of offsets (-1,000 to 300 ms in 100 ms intervals) that shifted the temporal relationship of neural activity relative to each feature.

**Regularization.** As many of the features are high-dimensional, we took several steps to counter overfitting. First, we used principal components analysis to decrease the effective dimensionality of our feature sets, retaining only the components required to explain 90% of the variance in the temporal windows for each feature. To further address overfitting, we used elastic net regularization with an L1 weight of 0.5 and an  $\alpha$  value of 0.01. Qualitatively, results were not sensitive to changes in these parameter choices.

**Cross validation.** We trained GLMs using a fivefold cross-validation scheme. We first divided the spiking, movement and ANN data spanning the duration of a unit's recording into 4 s chunks that were randomly distributed into ten folds. We trained GLMs using training data from nine of the folds and evaluated their performance on testing data from the remaining fold, training a single model for each combination of training and testing sets. We use the cross-validated log-likelihood ratio (CV-LLR) and deviance-ratio pseudo- $R^2$  to quantify model predictivity, the performance of a model in predicting spike counts in the testing set.

**Hypothesis testing.** We defined the most predictive feature for a given unit as the feature with the highest average CV-LLR. To identify units for which the features had low predictivity, we used a one-sided Wilcoxon signed-rank test to assess whether the CV-LLRs for each unit and each feature sufficiently deviated from zero. Units with a confidence score greater than 0.05 with Bonferroni correction for several comparisons were labelled as unclassified.

### Representational similarity analysis

We used RSA to compare the representational structure of neural activity in DLS and MC across behaviours to measurable features of movement, dynamic features inferred from physical simulation and the activations of ANN inverse controllers. RSA consists of three broad steps: feature vector estimation, RDM estimation and RDM comparison.

**Feature vector estimation.** We first applied principal components analysis to each feature, retaining only the components required to explain 95% of the total variance. For each session, we used the behavioural labels from our automated behavioural segmentation, applied a 200 ms iterative mode filter to mitigate short-duration bouts and divided samples from each feature into behavioural categories. To eventually achieve an unbiased estimate in the dissimilarity between behavioural categories for a given feature, we divided data into two partitions for each behaviour, with odd instances of the behaviour comprising the first partition and even instances of the behaviour

comprising the second partition. For each partition, we computed the average feature vector across all samples.

We excluded frames in which the animal was sleeping and frames in the 40 samples surrounding the initiation or termination of the model's imitation episodes. We only included sessions in which a minimum of 10 simultaneously recorded neurons were present throughout the entire duration of the session, sessions in which a minimum of 70% of the total behavioural categories were expressed and sessions in which there was a minimum of 30 min of movement.

**RDM estimation.** We used rsatoolbox 3.0 to perform RDM estimation using the cross-validated squared Mahalanobis distance (crossnobis dissimilarity)<sup>71-73</sup> with the feature vectors from the behavioural partitions described above. This produces a RDM for each feature and each session. Although the models' conditional latent bottlenecks naturally indicate calculation of RDMs using distance metrics for distributions, such as the symmetric Kullback-Liebler divergence, it was challenging to compare these metrics across features for which we do not have parametrized probability distributions. Thus, we chose to separately analyse the latent means and scales, as well as all other features, using crossnobis dissimilarity.

**RDM comparison.** For each feature, we computed the average RDM across sessions and compared RDMs across features and subjects using the whitened unbiased cosine similarity<sup>71,72</sup>.

### Motor variability analyses

**Estimating instantaneous motor variability.** We modified the normal inference procedure to estimate the instantaneous motor variability of the model at each timestep. Rather than disabling the latent variability, we generated 50 latent samples from the latent distribution at each frame. We then evaluated the decoder for each sample to estimate the distribution of actions that emerged from a given latent distribution. We use the standard deviation across the distribution of actions for each actuator as the instantaneous estimate of actuator variability.

To assess the significance of the predictivity of action variability and latent variability on behaviour, we performed a permutation test. For each of 1,000 iterations, we trained a logistic regression classifier using balanced class weights to predict the behavioural category from the vector of action standard deviations or latent standard deviations at each timepoint. We also trained another logistic regression classifier using randomly permuted category labels. The performance of both classifiers were evaluated with fivefold cross validation, using the class-balanced accuracy as a performance metric.

**Variability perturbations.** We further modified the inference procedure to perturb the structure of latent variability. Our perturbation involved varying the structure of the latent variability and clamping the total variability of the latent space at each timepoint. We considered three different structures for the latent variability. The first was a standard variability structure, in which no changes were made to the latent distribution. The second was a uniform variability structure in which each dimension of the latent space was set to equal variance for every frame. The third was an inverted variability structure that was constructed as follows. In each frame, the latent dimensions were ranked according to their latent standard deviation. The standard deviations were then reassigned in inverse rank order such that the dimensions with the highest variability were assigned low variability and vice versa. To clamp the variability at a particular noise level, we multiplied the transformed latent variability vector by a scalar value such that the total variability across all dimensions in the latent space equalled the desired noise level for every frame of the simulation.

To evaluate the performance of models undergoing variability perturbations, we defined a fictive reward term to combine the several

# Article

MO-VMPO objectives into a single scalar value. The fictive reward was adapted from objective functions in previous work<sup>3</sup> and was defined by

$$0.4r_{\text{com}} + 2r_{\text{vel}} + 0.15r_{\text{app}} + 0.35r_{\text{quat}}$$

**Estimating instantaneous neural variability.** Although the parametrizations of variability in latent variable models can be easily recorded, directly measuring instantaneous variability in neural activity is not possible. To approximate a measure of instantaneous neural variability, we computed a sliding-window variance estimate using a 1 s window on the binned spike counts of each neuron. In lieu of more sophisticated approaches that can estimate latent variability structure of neural populations across behaviour and at the months-long recording scale, we believe that our approach serves as a reasonable approximation of neural variability structure.

## Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

## Data availability

The data generated from real animals are publicly available on Harvard Dataverse, <https://doi.org/10.7910/DVN/FB0MZT>. To help us understand use, provide support, fulfil custom requests and encourage collaboration, we ask that users contact us when considering using this dataset. Because of their size, the data generated in simulation will be made available on reasonable request.

## Code availability

Code for all analyses will be made available from the corresponding authors on reasonable request. Repositories for skeletal registration (STAC), behavioural classification (motion-mapper) and inverse dynamic model inference are available at [https://github.com/diegoaldarondo/virtual\\_rodent](https://github.com/diegoaldarondo/virtual_rodent).

62. Dhawale, A. K. et al. Automated long-term recording and analysis of neural activity in behaving animals. *eLife* **6**, e27702 (2017).
63. Chung, J. E. et al. A fully automated approach to spike sorting. *Neuron* **95**, 1381–1394.e6 (2017).

64. Merel, J. et al. Hierarchical visuomotor control of humanoids. In *Proc. 7th International Conference on Learning Representations (ICLR, 2019)*.
65. Chentanez, N., Müller, M., Macklin, M., Makoviychuk, V. & Jeschke, S. Physics-based motion capture imitation with deep reinforcement learning. In *Proc. 11th Annual International Conference on Motion, Interaction, and Games 1–10 (ACM, 2018)*.
66. Abdolmaleki, A. et al. A distributional view on multi-objective policy optimization. In *Proc. 37th International Conference on Machine Learning (eds Daumé, H. & Singh, A.) 11–22 (PMLR, 2020)*.
67. Francis Song, H. et al. V-MPO: on-policy maximum a posteriori policy optimization for discrete and continuous control. In *Proc. 8th International Conference on Learning Representations (2020)*.
68. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *Proc. 3rd International Conference on Learning Representations (eds Bengio, Y. & LeCun, Y.) (2015)*.
69. Maas, A. L., Hannun, A. Y. & Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *Proc. 30th International Conference on Machine Learning (ICML) (2013)*.
70. Seabold, S. & Perktold, J. Statsmodels: econometric and statistical modeling with Python. In *Proc. 9th Python in Science Conference (eds van der Walt, S. & Millman, J.) 92–96 (SciPy, 2010)*; <https://doi.org/10.25080/majora-92bf1922-011>.
71. Diedrichsen, J. et al. Comparing representational geometries using whitened unbiased-distance-matrix similarity. Preprint at <https://arxiv.org/abs/2007.02789> (2020).
72. Schütt, H. H., Kipnis, A. D., Diedrichsen, J. & Kriegeskorte, N. Statistical inference on representational geometries. Preprint at <https://arxiv.org/abs/2112.09200> (2021).
73. Nili, H. et al. A toolbox for representational similarity analysis. *PLoS Comput. Biol.* **10**, e1003553 (2014).

**Acknowledgements** We thank M. Shad and the team at Harvard Research Computing for their technical support. We are grateful to S. Wolff, K. Hardcastle and J. Casas for their support with experimental procedures. We would also like to thank S. Escola for feedback on our manuscript. This work was supported by a National Institutes of Health D-SPAN Award (1F99NS125834-01A1) to D.A. and National Institutes of Health grants (nos. R01NS099323, R01GM136972) to B.P.Ö. The illustration of the rat in Fig. 1a was hand drawn by D.A. from a model licensed from Biosphera3D.

**Author contributions** D.A., J.M., J.D.M., G.W., M.B. and B.P.Ö. conceived the project idea. D.A., U.K. and A.G. carried out the experiments. D.A. and U.K. processed the data. J.M. and L.H. trained the inverse dynamics models. D.A., J.M., J.D.M. and Y.T. contributed to the biomechanical model. D.A. analysed the data. D.A., J.M., J.D.M., L.H., M.B. and B.P.Ö. contributed to the interpretation of the results. D.A., J.M., J.D.M. and B.P.Ö. wrote the manuscript with input from all authors.

**Competing interests** The authors declare no competing interests.

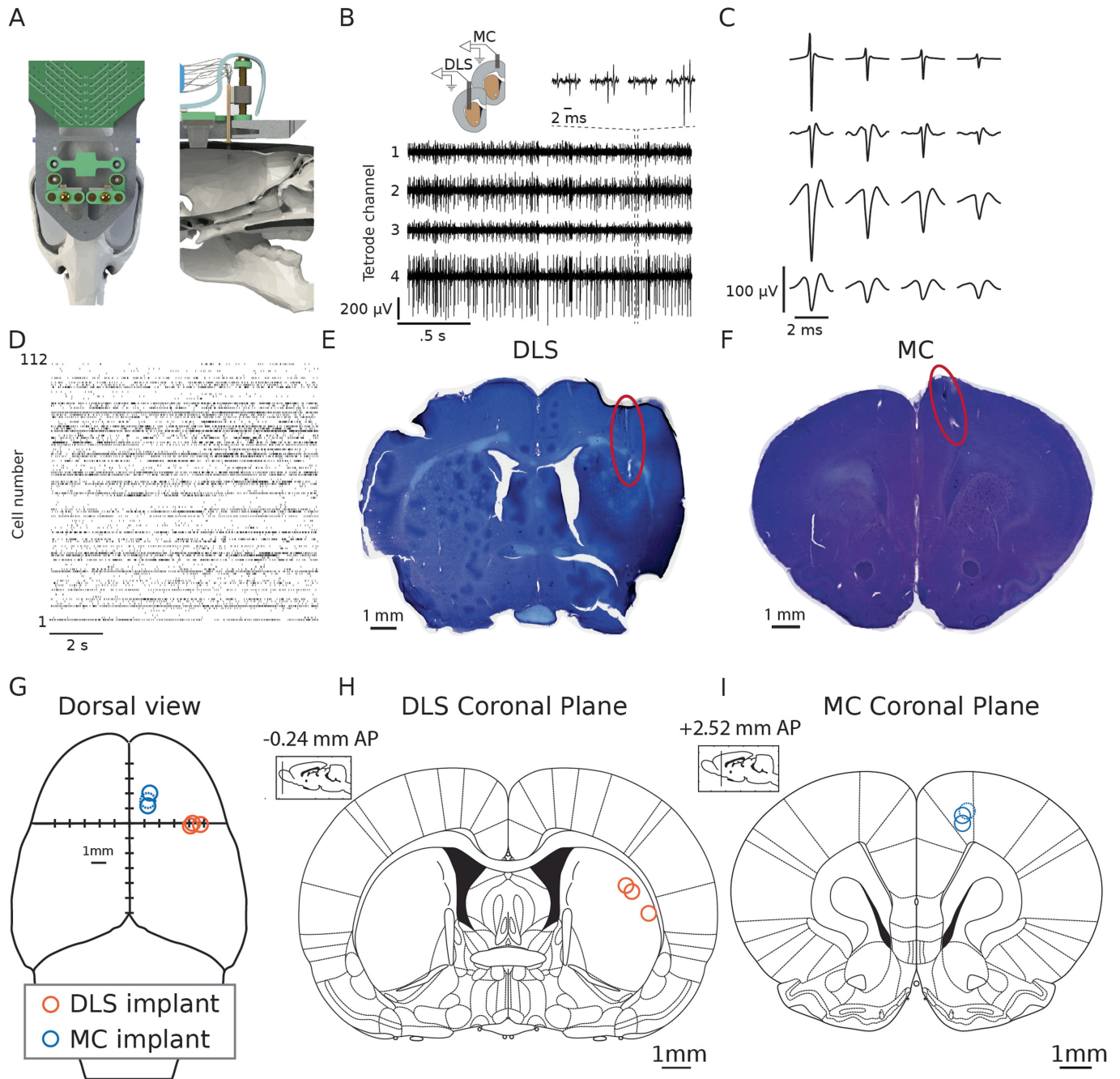
## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41586-024-07633-4>.

**Correspondence and requests for materials** should be addressed to Diego Aldarondo or Bence P. Ölveczky.

**Peer review information** Nature thanks Jonathan Whitlock and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>.

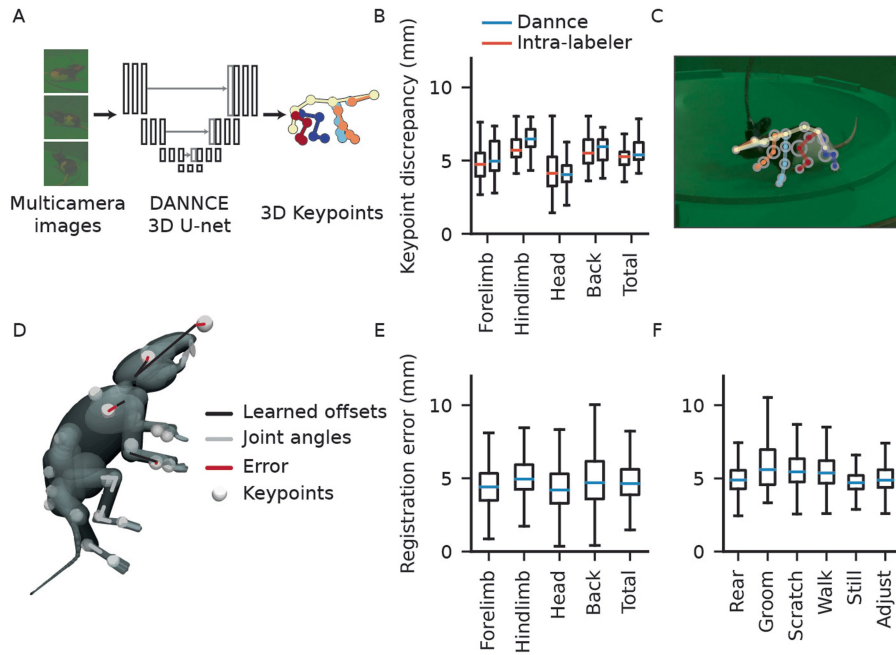


**Extended Data Fig. 1 | Recording neural activity in freely behaving rats.**

**A**) Schematic of custom 128-channel tetrode drive. **B**, Tetrodes record electrical events of several putative neurons from the DLS or MC. Shown are recordings from a tetrode in DLS. **C**) Individual putative cells are extracted based on their unique spike waveforms using a custom spike-sorting software, FAST. **D**) Tetrodes allows for the recording of hundreds of putative single units simultaneously. **E-F**) Representative examples of Nissl-stained brain slices from animals with electrophysiological implants in DLS and MC. Red ellipses indicate the lesions remaining from the tetrode implants. **G**, Dorsal view

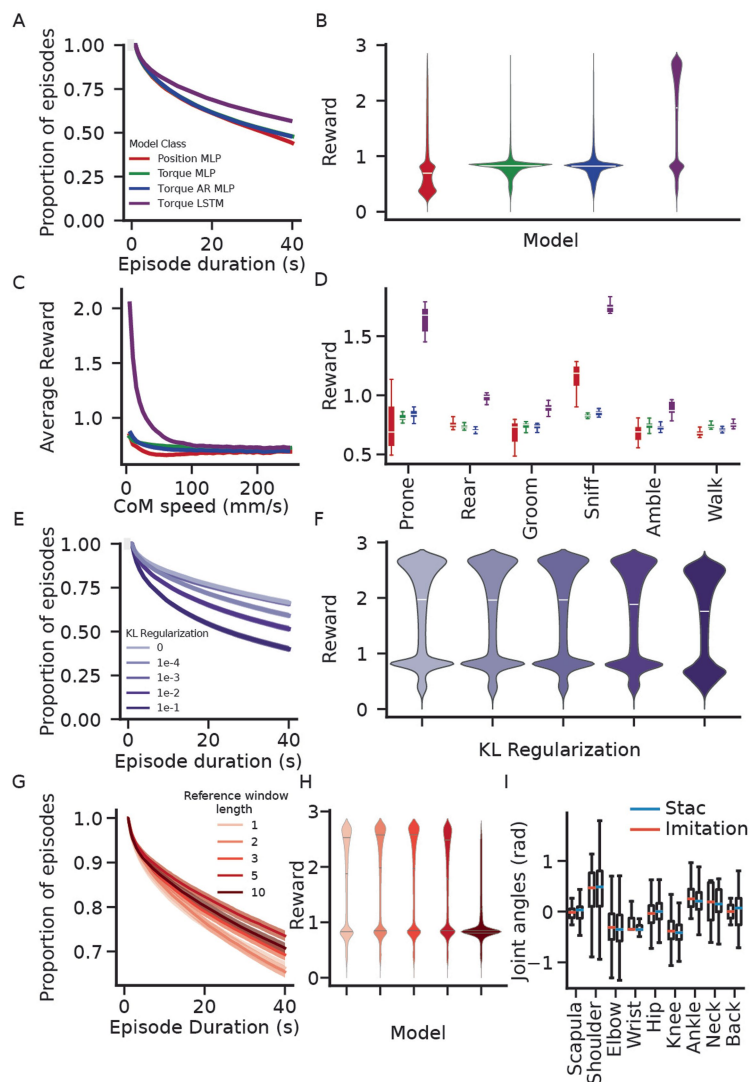
denoting the position of implants for DLS and MC. The position of the implant with the dashed circle could not be verified with histology as the recording headstage was dislodged prior to electric lesion. The position was instead estimated using scarring at the cortical surface and the recorded depth of implantation. The other implants were verified with electric lesions or scarring from the implant tip. **H**) Coronal plane indicating the location of implants in the DLS across 3 animals. **I**) Coronal plane indicating the location of implants in MC across 3 animals.





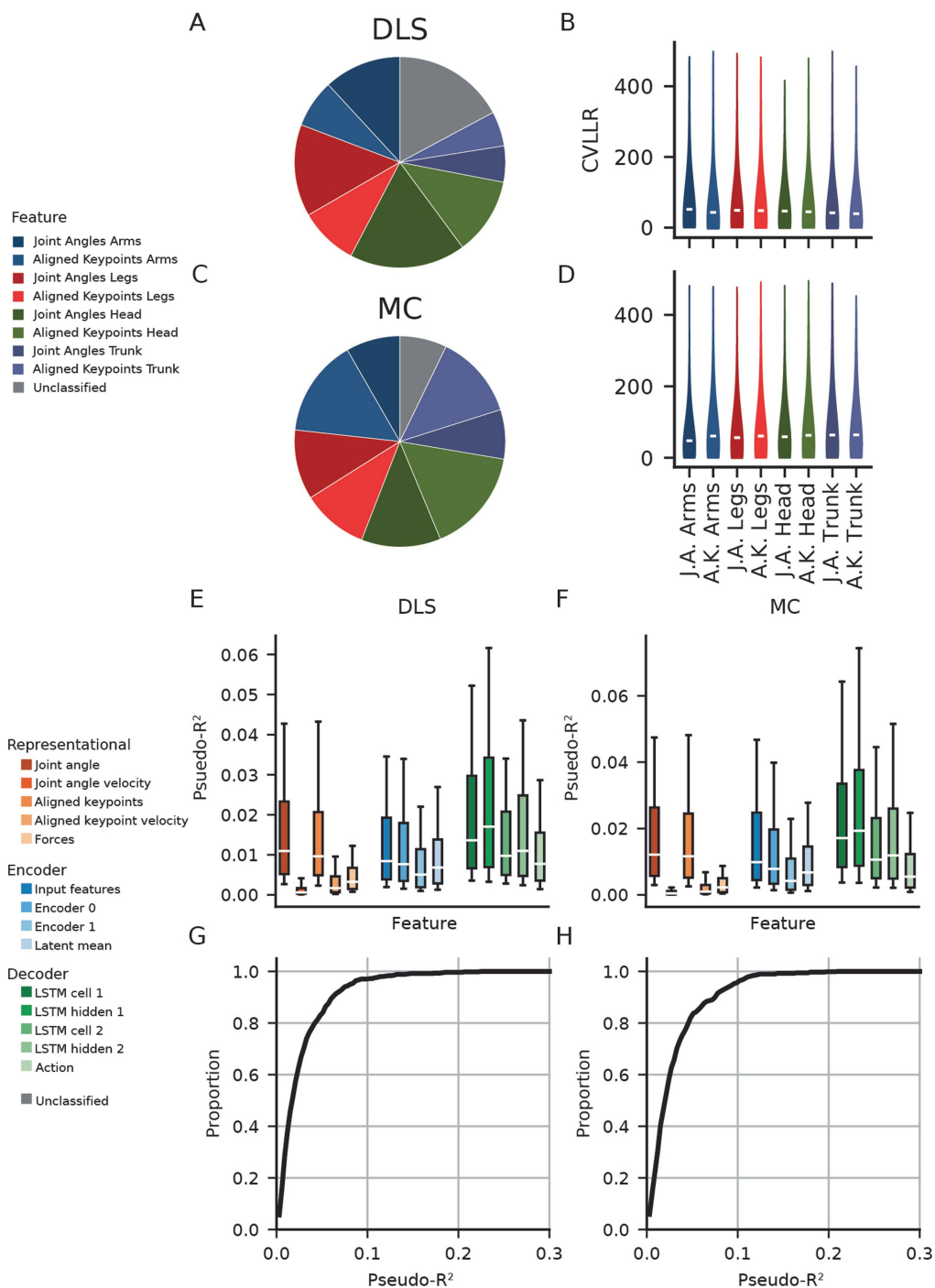
**Extended Data Fig. 2 | High fidelity 3D pose estimation and skeletal registration.** **A**) In DANNCE, a 3D U-Net processes multi-view images to estimate the positions of 23 3D keypoints across the rat's body. **B**, DANNCE keypoint estimates show high concordance with manual annotations, deviating from manual labels to a similar degree as repeated manual annotations of the same testing frames. **C**) Visualization of median DANNCE keypoint discrepancy relative to manual annotation. Grey circles indicate the bounds of the sphere

with radius equal to the median keypoint discrepancy for each keypoint. **D**) Schematic depicting the relevant variables in STAC. STAC operates by jointly optimizing a set of offsets relating the skeletal model to different keypoints and the pose of the model in each frame. **E**) STAC registration is highly accurate across body parts and **F**) across different behaviours. For all boxplots in this figure, coloured lines indicate the median, boxes indicate the interquartile range, and whiskers indicate the 10th and 90th percentiles.



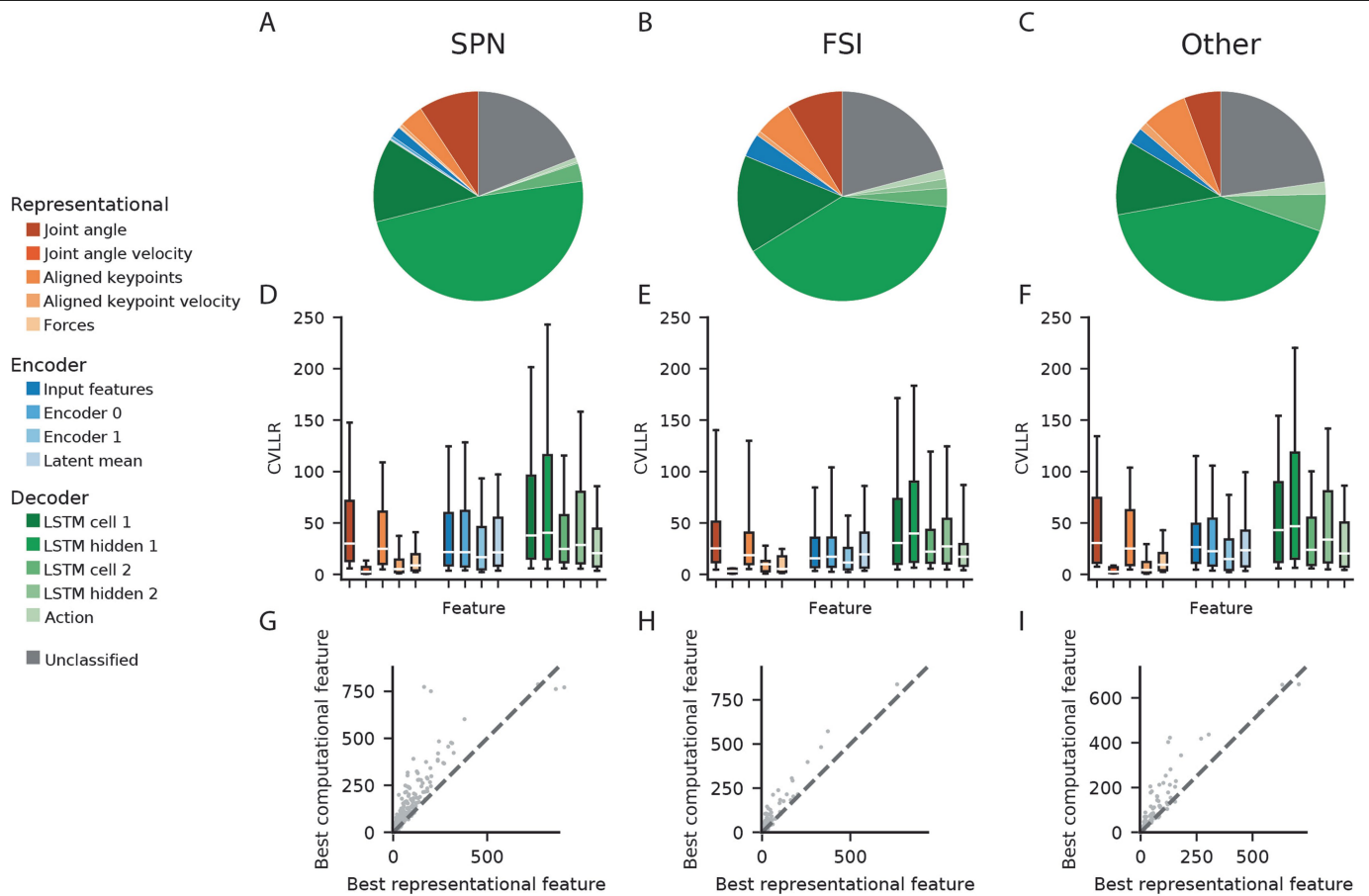
**Extended Data Fig. 3 | Comparing imitation performance for held-out data across different classes of control networks.** **A)** The proportion of episodes exceeding a given duration for the four classes of controllers. Results for each class are averaged across models with all KL regularization coefficients for that class. **B)** Violin plots showing the distribution of rewards by each model class on the held-out testing set. Models with LSTM decoders outperform other classes. **C)** Average reward as a function of the center of mass speed for each class of controller. LSTM models outperform other model classes across all speeds, but especially at slow speeds. **D)** Box plots denoting the distribution of rewards for each model class as a function of behavior category. LSTM models outperform other classes across all behavior, but especially those with slow center of mass speed. White lines indicate the median, box limits indicate the interquartile range, box whiskers indicate the 10th and 90th percentiles. **E)** The proportion of episodes exceeding a given duration for models with LSTM decoders across all KL regularization coefficients. Models with higher KL regularization are generally less robust than those with lower KL regularization, consistent with an increase in latent noise. **F)** Violin plots denoting the distribution of rewards on held-out natural behavior for each model as a function of KL regularization. Increasing the KL regularization coefficient marginally decreases the reward distribution of the models. White lines indicate the median. **G)** We trained five models with different reference window lengths using an LSTM decoder with a KL regularization of  $1e-4$ . Violin plots denote the distribution of rewards on held-out natural behavior for each model. Models with reference windows of length 5 or shorter exhibit comparable performance, while a reference window of 10 exhibits poorer performance. Grey lines indicate the quartiles. **H)** The proportion of episodes exceeding a given duration. Models with longer reference window length are generally more robust than those with shorter reference window lengths, with the most robust model being that with a reference window length of 5. Shaded regions indicate the standard error of the mean over sessions. **I)** The distribution of joint angles during imitation closely match those of STAC-registered skeletal models during imitation. Data is from a model with an LSTM decoder and a KL regularization of  $1e-4$ . Box centers indicate the median, box limits indicate the interquartile range, box whiskers indicate the maximum or minimum values up to 1.5 times the interquartile range from the box limits.

Increasing the KL regularization coefficient marginally decreases the reward distribution of the models. White lines indicate the median. **G)** We trained five models with different reference window lengths using an LSTM decoder with a KL regularization of  $1e-4$ . Violin plots denote the distribution of rewards on held-out natural behavior for each model. Models with reference windows of length 5 or shorter exhibit comparable performance, while a reference window of 10 exhibits poorer performance. Grey lines indicate the quartiles. **H)** The proportion of episodes exceeding a given duration. Models with longer reference window length are generally more robust than those with shorter reference window lengths, with the most robust model being that with a reference window length of 5. Shaded regions indicate the standard error of the mean over sessions. **I)** The distribution of joint angles during imitation closely match those of STAC-registered skeletal models during imitation. Data is from a model with an LSTM decoder and a KL regularization of  $1e-4$ . Box centers indicate the median, box limits indicate the interquartile range, box whiskers indicate the maximum or minimum values up to 1.5 times the interquartile range from the box limits.



**Extended Data Fig. 4 | Neurons in the DLS and MC encode posture across many body parts to a degree consistent with previous reports during unrestrained behavior.** **A, C)** Proportion of neurons in DLS and MC best predicted by each feature class. **B, D)** Violin plots showing the distribution of cross-validated log-likelihood ratios (CV-LLR) of GLMs trained to predict spike counts using different feature classes. **E, F)** Box plots showing the distribution of deviance-ratio pseudo-r-squared values of GLMs trained to predict spike

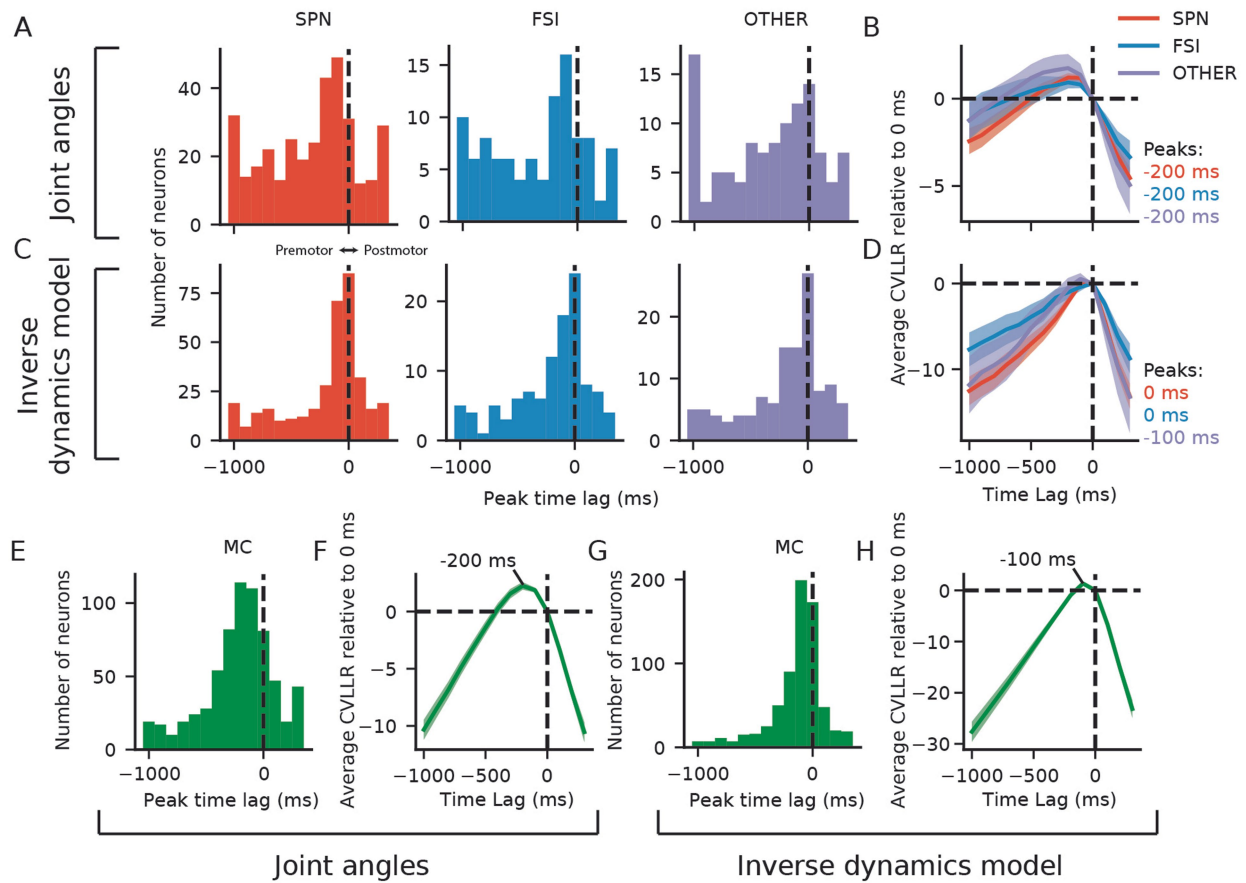
counts using different feature classes. White lines indicate the median, boxes indicate the interquartile range, and whiskers indicate the 10th and 90th percentiles. **G, H)** Empirical cumulative distribution functions denoting the proportion of neurons in DLS and MC with peak GLM predictivity below a given pseudo-r-squared value. The distributions resemble previous reports in rats during spontaneous behavior<sup>42</sup>.



**Extended Data Fig. 5 | Encoding properties are similar across striatal cell types.** **A-C)** Proportion of neurons in DLS and MC best predicted by each feature class for each cell type. **D-F)** Box plots showing the distribution of cross-validated log-likelihood ratios relative to a mean firing rate model for GLMs trained to predict spike counts using different feature classes. White lines indicate the median, boxes indicate the interquartile range, and whiskers

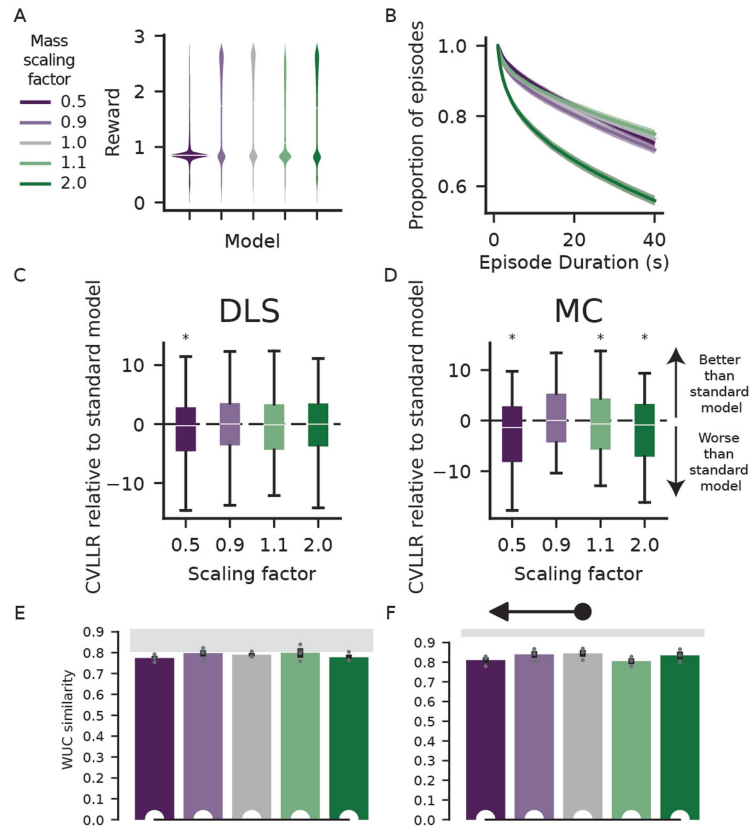
indicate the 10th and 90th percentiles. **G-H)** Comparison of the best computational feature derived from the network and representational feature GLM CV-LLRs for each neuron. GLMs based on the inverse dynamics models (computational features) outperform those based on representational features for the majority of classified neurons for all cell types ( $p < .001$ , permutation test).





**Extended Data Fig. 6 | Neurons in the DLS and MC encode future movement during natural behavior.** We trained GLMs to predict neural activity from measurable features of movement and from features of the ANN controllers while introducing time lags ranging from -1000 ms to 300 ms between neural activity and the features. **A)** Histograms depicting the distribution of time lags for maximally predictive GLMs when using joint angle predictors. Time lags less than zero correspond to neurons whose future movements better predict neural activity (premotor), while time lags greater than zero correspond to neurons whose past movements best predict neural activity (postmotor).

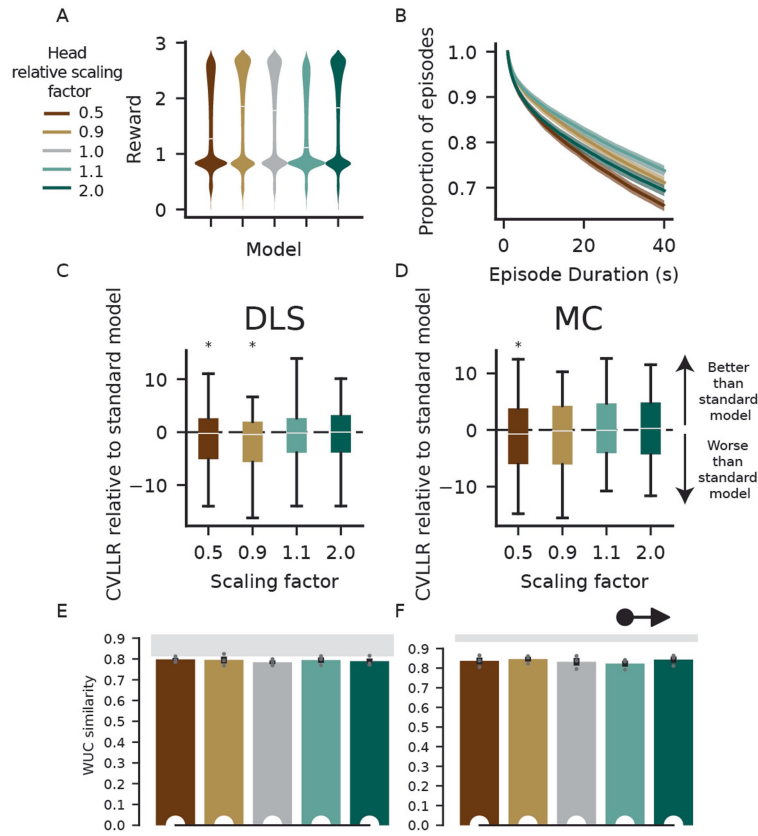
**B)** CVLLR relative to models trained with a time lag of 0 ms averaged across neurons. Shaded regions indicate the standard error of the mean. The peak average CVLLR occurs at -200 ms for all cell types. **C, D)** Same as A-B, except using features from the inverse dynamics model (LSTM hidden layer 1) as GLM predictors for a model with an LSTM decoder and a KL regularization of  $1e-4$ . Peak predictivity occurs closer to a time lag of zero, consistent with the network's representation of desired future state and inverse dynamics. **E, F)** Same as A-B for neurons in MC. **G, H)** Same as C-D for neurons in MC.



**Extended Data Fig. 7 | Comparing imitation performance and neural predictivity of models trained to control bodies of different masses.**

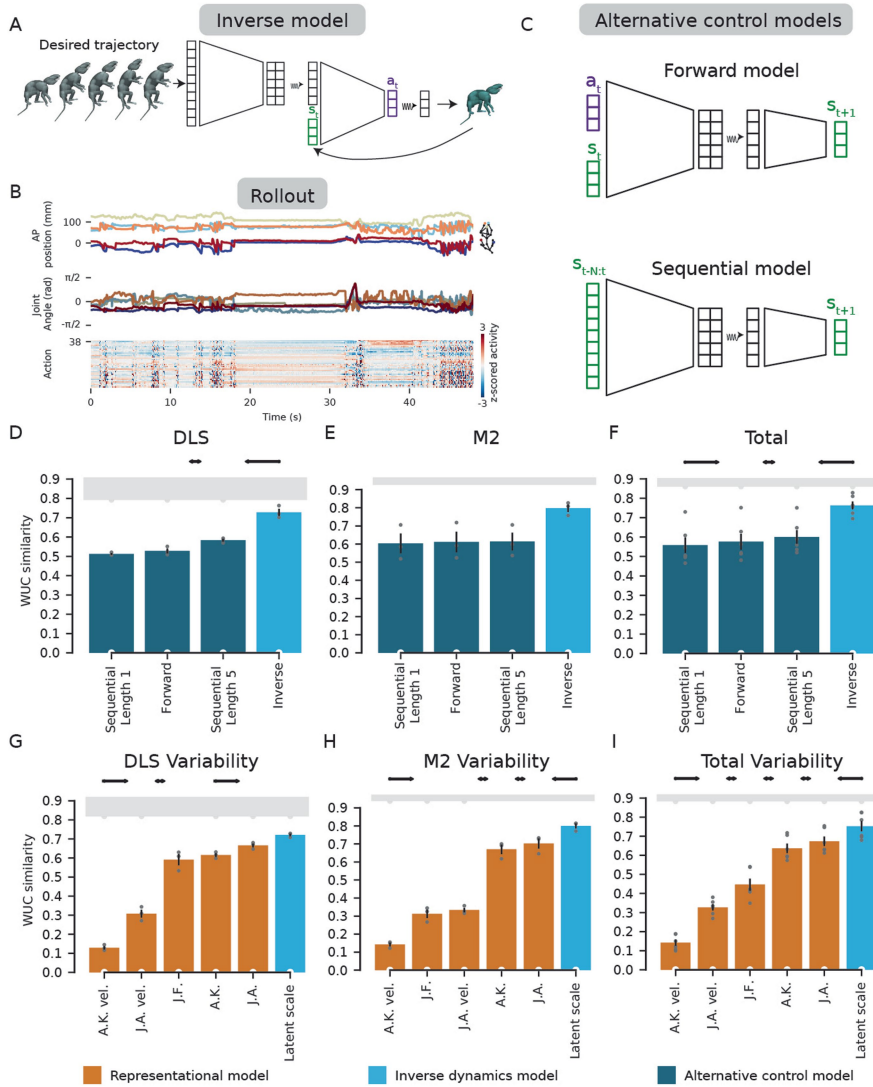
**A)** We trained five models with an LSTM decoder and a KL regularization of  $1e-4$  to control bodies of different masses. Violin plots denote the distribution of rewards on held-out natural behavior for each model. Several models controlling bodies with masses other than the standard mass exhibited reduced performance. White lines indicate medians. **B)** The proportion of episodes exceeding a given duration. Shaded regions indicate S.E.M across individuals. **C-D)** Box plots depicting the distribution of cross-validated log-likelihood ratios across neurons of GLMs trained to predict neural activity from network features. The CVLLR for each neuron is expressed relative to the likelihood of a GLM trained to predict neural activity using network features

from the standard mass model. Values greater than zero imply a model more predictive of neural activity than those derived from the standard mass model, and vice versa. White lines indicate the median, box limits indicate the quartiles, whiskers indicate the 10th and 90th percentiles. Stars indicate that a greater proportion of neurons are better predicted by GLMs trained using features from the standard mass model than from the alternative mass model (Bonferroni corrected,  $\alpha = .05$ , permutation test). **E-F)** Average WUC similarity between RDMs derived from network layers and neural activity in DLS or MC. Error bars indicate S.E.M across individuals. Arrows indicate significantly different similarity distributions across animals (Benjamini-Hochberg corrected, false discovery rate  $\alpha = .05$ , one-sided t-test).



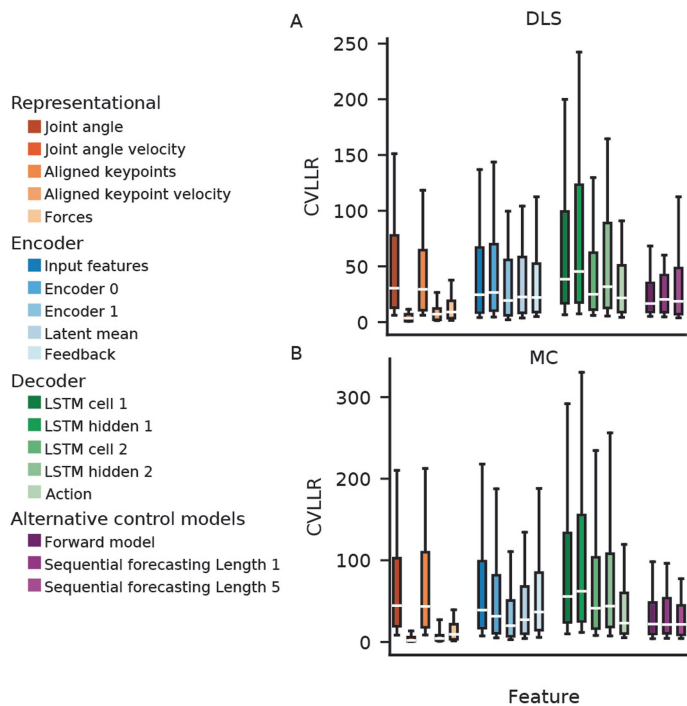
**Extended Data Fig. 8 | Comparing imitation performance and neural predictivity of models trained to control bodies of the same total mass with different head masses.** **A)** We trained five models with an LSTM decoder and a KL regularization of  $1e-4$  to control bodies of the same total mass with different relative masses between the head and the rest of the body. Violin plots denote the distribution of rewards on held-out natural behavior for each model. Several models controlling bodies with masses other than the standard mass exhibited reduced performance. White lines indicate medians. **B)** The proportion of episodes exceeding a given duration. Shaded regions indicate S.E.M across individuals. **C-D)** Box plots depicting the distribution of cross-validated log-likelihood ratios across neurons of GLMs trained to predict neural activity from network features. The CVLLR for each neuron is expressed relative to the

likelihood of a GLM trained to predict neural activity using network features from the standard mass model. Values greater than zero imply a model more predictive of neural activity than those derived from the standard mass model, and vice versa. White lines indicate the median, box limits indicate the quartiles, whiskers indicate the 10th and 90th percentiles. Stars indicate that a greater proportion of neurons are better predicted by GLMs trained using features from the standard mass model than from the alternative mass model (Bonferroni corrected,  $\alpha = .05$ , permutation test). **E-F)** Average WUC similarity between RDMs derived from network layers and neural activity in DLS or MC. Error bars indicate S.E.M across individuals. Arrows indicate significantly different similarity distributions across animals (Benjamini-Hochberg corrected, false discovery rate  $\alpha = .05$ , one-sided t-test).



**Extended Data Fig. 9 | The representational structures of DLS and MC resemble an inverse model more than alternative control models.** **A)** To compare the representational structure of neural activity in DLS and MC across different candidate computational models we used **B)** rollouts from an inverse model to collect state-action pairs to train **C)** forward and sequential models with supervised learning. **D-F)** Across-subject representational similarity between control models and neural activity. The latent representation of an inverse model more closely resembles the structure of neural activity in DLS

and MC than the latent representation of forward or sequential models. **G-I)** The latent variability of an inverse model better predicts the structure of neural variability than representational models. Error bars indicate S.E.M. Icicles and dew drops indicate significant differences from the noise ceiling and zero (Bonferroni corrected,  $\alpha = .05$ , one-sided t-test). Gray bars indicate the estimated noise ceiling of the true model. Arrows indicate significant differences between features (Benjamini-Hochberg corrected, false discovery rate  $\alpha = .05$ , one-sided t-test). Points indicate individual animals.



**Extended Data Fig. 10 | Inverse dynamics models predict putative single-unit neural activity better than alternative control models and feedback.** **A-B)** Box plots showing the distribution of cross-validated log-likelihood ratios (CV-LLR) relative to mean firing-rate models of GLMs trained to predict spike counts using different feature classes. White lines indicate the median, boxes indicate the interquartile range, and whiskers indicate the 10th and 90th percentiles.



## Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our [Editorial Policies](#) and the [Editorial Policy Checklist](#).

### Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a Confirmed

- The exact sample size ( $n$ ) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided  
*Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g.  $F$ ,  $t$ ,  $r$ ) with confidence intervals, effect sizes, degrees of freedom and  $P$  value noted  
*Give  $P$  values as exact values whenever suitable.*
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen's  $d$ , Pearson's  $r$ ), indicating how they were calculated

*Our web collection on [statistics for biologists](#) contains articles on many of the points above.*

### Software and code

Policy information about [availability of computer code](#)

Data collection

Multi-camera video was acquired using Campy (<https://github.com/ksseverson57/campy>). Camera calibration was performed using the MATLAB Camera Calibration App (Mathworks) and opencv-python. Electrophysiological data was acquired using RHD2000 interface (Intan).

Data analysis

We estimated the 3D pose from multi-camera videos using DANNCE (version 1.3) (Dunn et al. 2021).

Skeletal registration was performed using a custom implementation of simultaneous tracking and calibration (STAC) (Wu et al. 2013).

Inverse dynamics model training was performed using CoMic (Hasenclever et al. 2020).

Inverse dynamic model inference was performed using a custom python package, npmp\_embedding, which depends on dm\_control, tensorflow (version 2.3), and MuJoCo (version 2.10).

Behavioral classification was performed using a custom MATLAB implementation of motion-mapper (Berman et al. 2014).

Spike sorting was performed using Ephys2, a modification of FAST (Dhawale et al. 2017).

All data analyses were performed with custom code written in Python using packages from the PyData stack (NumPy, SciPy, StatsModels, Sci-kit learn, matplotlib), as well as rsatoolbox 3.0 for representational similarity analyses. Alternative control models were trained using custom python code dependent on Tensorflow 2.8.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio [guidelines for submitting code & software](#) for further information.

## Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our [policy](#)

The generated datasets are available from the corresponding author upon reasonable request.

## Human research participants

Policy information about [studies involving human research participants and Sex and Gender in Research](#).

Reporting on sex and gender	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

## Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

- Life sciences       Behavioural & social sciences       Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see [nature.com/documents/nr-reporting-summary-flat.pdf](https://www.nature.com/documents/nr-reporting-summary-flat.pdf)

## Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	No statistical methods were used to pre-determine the number of subjects in our study but our sample sizes for electrophysiological data are similar to those reported in previous publications (Jin et al., 2014; Panigrahi et al., 2015; Rueda-Orozco and Robbe, 2015; Dhawale & Wolff 2021). These include N=3 animals with implants in the sensorimotor striatum, and N=4 animals with implants in the motor cortex.
Data exclusions	No animals were excluded from behavioral analyses. One of the implanted animals (motor cortex) yielded a recording in which no neurons could be detected, and thus this animal was excluded from electrophysiological analyses.
Replication	All recordings were performed with multiple animals per group (n=3) and yielded consistent, reproducible findings.
Randomization	Animals were assigned to groups for implantation in the striatum or motor cortex using a computer-generated random number list. This ensured an unbiased distribution and balanced potential covariates like litter order, without directly controlling for each covariate.
Blinding	Blinding was not feasible in our study because individual rats could be easily identified by distinctive features such as fur pattern, size, and behavior. Given personnel constraints, the same experimenter conducted most of the surgical implantations and behavioral experiments, making it impossible to blind them to the implantation sites. To minimize bias, we employed standardized and automated procedures for data collection and analysis where applicable.

## Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description	Briefly describe the study type including whether data are quantitative, qualitative, or mixed-methods (e.g. qualitative cross-sectional, quantitative experimental, mixed-methods case study).
Research sample	State the research sample (e.g. Harvard university undergraduates, villagers in rural India) and provide relevant demographic

Research sample	<i>information (e.g. age, sex) and indicate whether the sample is representative. Provide a rationale for the study sample chosen. For studies involving existing datasets, please describe the dataset and source.</i>
Sampling strategy	<i>Describe the sampling procedure (e.g. random, snowball, stratified, convenience). Describe the statistical methods that were used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient. For qualitative data, please indicate whether data saturation was considered, and what criteria were used to decide that no further sampling was needed.</i>
Data collection	<i>Provide details about the data collection procedure, including the instruments or devices used to record the data (e.g. pen and paper, computer, eye tracker, video or audio equipment) whether anyone was present besides the participant(s) and the researcher, and whether the researcher was blind to experimental condition and/or the study hypothesis during data collection.</i>
Timing	<i>Indicate the start and stop dates of data collection. If there is a gap between collection periods, state the dates for each sample cohort.</i>
Data exclusions	<i>If no data were excluded from the analyses, state so OR if data were excluded, provide the exact number of exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.</i>
Non-participation	<i>State how many participants dropped out/declined participation and the reason(s) given OR provide response rate OR state that no participants dropped out/declined participation.</i>
Randomization	<i>If participants were not allocated into experimental groups, state so OR describe how participants were allocated to groups, and if allocation was not random, describe how covariates were controlled.</i>

## Ecological, evolutionary & environmental sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description	<i>Briefly describe the study. For quantitative data include treatment factors and interactions, design structure (e.g. factorial, nested, hierarchical), nature and number of experimental units and replicates.</i>
Research sample	<i>Describe the research sample (e.g. a group of tagged <i>Passer domesticus</i>, all <i>Stenocereus thurberi</i> within Organ Pipe Cactus National Monument), and provide a rationale for the sample choice. When relevant, describe the organism taxa, source, sex, age range and any manipulations. State what population the sample is meant to represent when applicable. For studies involving existing datasets, describe the data and its source.</i>
Sampling strategy	<i>Note the sampling procedure. Describe the statistical methods that were used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient.</i>
Data collection	<i>Describe the data collection procedure, including who recorded the data and how.</i>
Timing and spatial scale	<i>Indicate the start and stop dates of data collection, noting the frequency and periodicity of sampling and providing a rationale for these choices. If there is a gap between collection periods, state the dates for each sample cohort. Specify the spatial scale from which the data are taken</i>
Data exclusions	<i>If no data were excluded from the analyses, state so OR if data were excluded, describe the exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.</i>
Reproducibility	<i>Describe the measures taken to verify the reproducibility of experimental findings. For each experiment, note whether any attempts to repeat the experiment failed OR state that all attempts to repeat the experiment were successful.</i>
Randomization	<i>Describe how samples/organisms/participants were allocated into groups. If allocation was not random, describe how covariates were controlled. If this is not relevant to your study, explain why.</i>
Blinding	<i>Describe the extent of blinding used during data acquisition and analysis. If blinding was not possible, describe why OR explain why blinding was not relevant to your study.</i>

Did the study involve field work?  Yes  No

## Field work, collection and transport

Field conditions	<i>Describe the study conditions for field work, providing relevant parameters (e.g. temperature, rainfall).</i>
Location	<i>State the location of the sampling or experiment, providing relevant parameters (e.g. latitude and longitude, elevation, water depth).</i>
Access & import/export	<i>Describe the efforts you have made to access habitats and to collect and import/export your samples in a responsible manner and in</i>

Access & import/export	<i>compliance with local, national and international laws, noting any permits that were obtained (give the name of the issuing authority, the date of issue, and any identifying information).</i>
Disturbance	<i>Describe any disturbance caused by the study and how it was minimized.</i>

## Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

### Materials & experimental systems

n/a	Involved in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> Antibodies
<input checked="" type="checkbox"/>	<input type="checkbox"/> Eukaryotic cell lines
<input checked="" type="checkbox"/>	<input type="checkbox"/> Palaeontology and archaeology
<input type="checkbox"/>	<input checked="" type="checkbox"/> Animals and other organisms
<input checked="" type="checkbox"/>	<input type="checkbox"/> Clinical data
<input checked="" type="checkbox"/>	<input type="checkbox"/> Dual use research of concern

### Methods

n/a	Involved in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> ChIP-seq
<input checked="" type="checkbox"/>	<input type="checkbox"/> Flow cytometry
<input checked="" type="checkbox"/>	<input type="checkbox"/> MRI-based neuroimaging

## Antibodies

Antibodies used	<i>Describe all antibodies used in the study; as applicable, provide supplier name, catalog number, clone name, and lot number.</i>
Validation	<i>Describe the validation of each primary antibody for the species and application, noting any validation statements on the manufacturer's website, relevant citations, antibody profiles in online databases, or data provided in the manuscript.</i>

## Eukaryotic cell lines

Policy information about [cell lines and Sex and Gender in Research](#)

Cell line source(s)	<i>State the source of each cell line used and the sex of all primary cell lines and cells derived from human participants or vertebrate models.</i>
Authentication	<i>Describe the authentication procedures for each cell line used OR declare that none of the cell lines used were authenticated.</i>
Mycoplasma contamination	<i>Confirm that all cell lines tested negative for mycoplasma contamination OR describe the results of the testing for mycoplasma contamination OR declare that the cell lines were not tested for mycoplasma contamination.</i>
Commonly misidentified lines (See <a href="#">ICLAC</a> register)	<i>Name any commonly misidentified cell lines used in the study and provide a rationale for their use.</i>

## Palaeontology and Archaeology

Specimen provenance	<i>Provide provenance information for specimens and describe permits that were obtained for the work (including the name of the issuing authority, the date of issue, and any identifying information). Permits should encompass collection and, where applicable, export.</i>
Specimen deposition	<i>Indicate where the specimens have been deposited to permit free access by other researchers.</i>
Dating methods	<i>If new dates are provided, describe how they were obtained (e.g. collection, storage, sample pretreatment and measurement), where they were obtained (i.e. lab name), the calibration program and the protocol for quality assurance OR state that no new dates are provided.</i>
<input type="checkbox"/>	Tick this box to confirm that the raw and calibrated dates are available in the paper or in Supplementary Information.
Ethics oversight	<i>Identify the organization(s) that approved or provided guidance on the study protocol, OR state that no ethical approval or guidance was required and explain why not.</i>

Note that full information on the approval of the study protocol must also be provided in the manuscript.

## Animals and other research organisms

Policy information about [studies involving animals](#); [ARRIVE guidelines](#) recommended for reporting animal research, and [Sex and Gender in Research](#)

Laboratory animals	Experimental subjects were female Long Evans rats (RRID: RGD_2308852) 3-12 months old at the start of recording.
Wild animals	The study did not involve wild animals.
Reporting on sex	The study only used female rats
Field-collected samples	The study did not contain samples collected from the field.
Ethics oversight	The care and experimental manipulation of all animals were reviewed and approved by the Harvard Institutional Animal Care and Use Committee.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

## Clinical data

Policy information about [clinical studies](#)

All manuscripts should comply with the ICMJE [guidelines for publication of clinical research](#) and a completed [CONSORT checklist](#) must be included with all submissions.

Clinical trial registration	<i>Provide the trial registration number from ClinicalTrials.gov or an equivalent agency.</i>
Study protocol	<i>Note where the full trial protocol can be accessed OR if not available, explain why.</i>
Data collection	<i>Describe the settings and locales of data collection, noting the time periods of recruitment and data collection.</i>
Outcomes	<i>Describe how you pre-defined primary and secondary outcome measures and how you assessed these measures.</i>

## Dual use research of concern

Policy information about [dual use research of concern](#)

### Hazards

Could the accidental, deliberate or reckless misuse of agents or technologies generated in the work, or the application of information presented in the manuscript, pose a threat to:

No	Yes	
<input type="checkbox"/>	<input type="checkbox"/>	Public health
<input type="checkbox"/>	<input type="checkbox"/>	National security
<input type="checkbox"/>	<input type="checkbox"/>	Crops and/or livestock
<input type="checkbox"/>	<input type="checkbox"/>	Ecosystems
<input type="checkbox"/>	<input type="checkbox"/>	Any other significant area

### Experiments of concern

Does the work involve any of these experiments of concern:

No	Yes	
<input type="checkbox"/>	<input type="checkbox"/>	Demonstrate how to render a vaccine ineffective
<input type="checkbox"/>	<input type="checkbox"/>	Confer resistance to therapeutically useful antibiotics or antiviral agents
<input type="checkbox"/>	<input type="checkbox"/>	Enhance the virulence of a pathogen or render a nonpathogen virulent
<input type="checkbox"/>	<input type="checkbox"/>	Increase transmissibility of a pathogen
<input type="checkbox"/>	<input type="checkbox"/>	Alter the host range of a pathogen
<input type="checkbox"/>	<input type="checkbox"/>	Enable evasion of diagnostic/detection modalities
<input type="checkbox"/>	<input type="checkbox"/>	Enable the weaponization of a biological agent or toxin
<input type="checkbox"/>	<input type="checkbox"/>	Any other potentially harmful combination of experiments and agents



## ChIP-seq

### Data deposition

- Confirm that both raw and final processed data have been deposited in a public database such as [GEO](#).
- Confirm that you have deposited or provided access to graph files (e.g. BED files) for the called peaks.

#### Data access links

May remain private before publication.

For "Initial submission" or "Revised version" documents, provide reviewer access links. For your "Final submission" document, provide a link to the deposited data.

#### Files in database submission

Provide a list of all files available in the database submission.

#### Genome browser session

(e.g. [UCSC](#))

Provide a link to an anonymized genome browser session for "Initial submission" and "Revised version" documents only, to enable peer review. Write "no longer applicable" for "Final submission" documents.

### Methodology

#### Replicates

Describe the experimental replicates, specifying number, type and replicate agreement.

#### Sequencing depth

Describe the sequencing depth for each experiment, providing the total number of reads, uniquely mapped reads, length of reads and whether they were paired- or single-end.

#### Antibodies

Describe the antibodies used for the ChIP-seq experiments; as applicable, provide supplier name, catalog number, clone name, and lot number.

#### Peak calling parameters

Specify the command line program and parameters used for read mapping and peak calling, including the ChIP, control and index files used.

#### Data quality

Describe the methods used to ensure data quality in full detail, including how many peaks are at FDR 5% and above 5-fold enrichment.

#### Software

Describe the software used to collect and analyze the ChIP-seq data. For custom code that has been deposited into a community repository, provide accession details.

## Flow Cytometry

### Plots

Confirm that:

- The axis labels state the marker and fluorochrome used (e.g. CD4-FITC).
- The axis scales are clearly visible. Include numbers along axes only for bottom left plot of group (a 'group' is an analysis of identical markers).
- All plots are contour plots with outliers or pseudocolor plots.
- A numerical value for number of cells or percentage (with statistics) is provided.

### Methodology

#### Sample preparation

Describe the sample preparation, detailing the biological source of the cells and any tissue processing steps used.

#### Instrument

Identify the instrument used for data collection, specifying make and model number.

#### Software

Describe the software used to collect and analyze the flow cytometry data. For custom code that has been deposited into a community repository, provide accession details.

#### Cell population abundance

Describe the abundance of the relevant cell populations within post-sort fractions, providing details on the purity of the samples and how it was determined.

#### Gating strategy

Describe the gating strategy used for all relevant experiments, specifying the preliminary FSC/SSC gates of the starting cell population, indicating where boundaries between "positive" and "negative" staining cell populations are defined.

- Tick this box to confirm that a figure exemplifying the gating strategy is provided in the Supplementary Information.

## Magnetic resonance imaging

### Experimental design

#### Design type

Indicate task or resting state; event-related or block design.

Design specifications

Specify the number of blocks, trials or experimental units per session and/or subject, and specify the length of each trial or block (if trials are blocked) and interval between trials.

Behavioral performance measures

State number and/or type of variables recorded (e.g. correct button press, response time) and what statistics were used to establish that the subjects were performing the task as expected (e.g. mean, range, and/or standard deviation across subjects).

## Acquisition

Imaging type(s)

Specify: functional, structural, diffusion, perfusion.

Field strength

Specify in Tesla

Sequence &amp; imaging parameters

Specify the pulse sequence type (gradient echo, spin echo, etc.), imaging type (EPI, spiral, etc.), field of view, matrix size, slice thickness, orientation and TE/TR/flip angle.

Area of acquisition

State whether a whole brain scan was used OR define the area of acquisition, describing how the region was determined.

Diffusion MRI

 Used Not used

## Preprocessing

Preprocessing software

Provide detail on software version and revision number and on specific parameters (model/functions, brain extraction, segmentation, smoothing kernel size, etc.).

Normalization

If data were normalized/standardized, describe the approach(es): specify linear or non-linear and define image types used for transformation OR indicate that data were not normalized and explain rationale for lack of normalization.

Normalization template

Describe the template used for normalization/transformation, specifying subject space or group standardized space (e.g. original Talairach, MNI305, ICBM152) OR indicate that the data were not normalized.

Noise and artifact removal

Describe your procedure(s) for artifact and structured noise removal, specifying motion parameters, tissue signals and physiological signals (heart rate, respiration).

Volume censoring

Define your software and/or method and criteria for volume censoring, and state the extent of such censoring.

## Statistical modeling & inference

Model type and settings

Specify type (mass univariate, multivariate, RSA, predictive, etc.) and describe essential details of the model at the first and second levels (e.g. fixed, random or mixed effects; drift or auto-correlation).

Effect(s) tested

Define precise effect in terms of the task or stimulus conditions instead of psychological concepts and indicate whether ANOVA or factorial designs were used.

Specify type of analysis:

Whole brain

ROI-based

Both

Statistic type for inference  
(See [Eklund et al. 2016](#))

Specify voxel-wise or cluster-wise and report all relevant parameters for cluster-wise methods.

Correction

Describe the type of correction and how it is obtained for multiple comparisons (e.g. FWE, FDR, permutation or Monte Carlo).

## Models & analysis

n/a | Involved in the study

  Functional and/or effective connectivity  Graph analysis  Multivariate modeling or predictive analysis

Functional and/or effective connectivity

Report the measures of dependence used and the model details (e.g. Pearson correlation, partial correlation, mutual information).

Graph analysis

Report the dependent variable and connectivity measure, specifying weighted graph or binarized graph, subject- or group-level, and the global and/or node summaries used (e.g. clustering coefficient, efficiency, etc.).

Multivariate modeling and predictive analysis

Specify independent variables, features extraction and dimension reduction, model, training and evaluation metrics.